# Bitcoin:
# Programming the Future of Money

Topics in Computer Science - ITCS 4010/5010, Spring 2025

Dr. Christian Kümmerle

## Lecture 7

## Bitcoin History, Hash Pointers

Some slides are adapted from:
- Coursera MOOC “Bitcoin and Cryptocurrency Technologies” (Bonneau, Felten, Narayanan and Miller)
- “Mastering Bitcoin: Programming the Open Blockchain”, (Andreas Antonopoulos, David Harding), 3rd Edition, O’Reilly, 2023.

# Basics of the Bitcoin Protocol

# BITCOIN: HOW IT STARTED

- **October 31, 2008**: Satoshi Nakamoto announces “Bitcoin: A Peer-to-Peer Electronic Cash System” to a cryptography mailing list
- **January 8, 2009**: S.N. releases first version of the Bitcoin node software, first Bitcoin transaction block has message embedded:
  **Times 03/Jan/2009 Chancellor on brink of second bailout for banks**

# BITCOIN: HOW IT STARTED

```
From:     Satoshi Nakamoto
Subject:  Bitcoin P2P e-cash paper
Date:     October 31, 2008 at 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.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf

**The main properties:**

- **Double-spending** is **prevented with a peer-to-peer network**.
- **No** mint or other **trusted parties**.
- Participants can be **anonymous.**
- **New coins** are made from **Hashcash style proof-of-work.**
- The proof-of-work for new coin generation **also powers the network to prevent double-spending.**

Source: https://satoshi.nakamotoinstitute.org/emails/cryptography/1/

# BITCOIN: THE WHITEPAPER

## Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

**Abstract.** A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

### 1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot

Published on
October 31, 2008

# BITCOIN: HOW IT STARTED

**Cryptography Mailing List** #014814

From: James A. Donald
Subject: **Bitcoin P2P e-cash paper**
Date: November 2, 2008 at 23:46:23 UTC

We very, very much need such a system, **but the way I understand your proposal, it does not seem to scale to the required size**.

For transferable proof of work tokens to have value, they must have monetary value. To have monetary value, they must be transferred within a very large network - for example a file trading network akin to bittorrent.

To **detect and reject a double spending event in a timely manner, one must have most past transactions of the coins in the transaction**, which, naively implemented, requires **each peer to have most past transactions**, or most past transactions that occurred recently. **If hundreds of millions of people are doing transactions, that is a lot of bandwidth** - each must know all, or a substantial part thereof.

Source: https://satoshi.nakamotoinstitute.org/emails/cryptography/threads/1/#014810

# BITCOIN: HOW IT STARTED

**Cryptography Mailing List** #014822

From: Ray Dillinger
Subject: **Bitcoin P2P e-cash paper**
Date: November 6, 2008 at 05:14:37 UTC

I think the real issue with this system is the market for bitcoins.

Computing proofs-of-work have no intrinsic value. We can have a limited supply curve (although the "currency" is inflationary at about 35% as that's how much faster computers get annually) but there is no demand curve that intersects it at a positive price point.

Source: https://satoshi.nakamotoinstitute.org/emails/cryptography/threads/1/#014822

# BITCOIN: HOW IT STARTED

- **October 31, 2008**: Satoshi Nakamoto announces “Bitcoin: A Peer-to-Peer Electronic Cash System” to a cryptography mailing list
- **January 8, 2009**: S.N. releases first version of the Bitcoin node software, first Bitcoin transaction block has message embedded:
  **Times 03/Jan/2009 Chancellor on brink of second bailout for banks**

- May 22, 2010:  Laszlo Hanyecz buys two pizzas with 10,000 bitcoins
- December 2010: Last forum post of Satoshi Nakamoto
- February 2011: 1 bitcoin is worth $1 for the first time
- February 2014: Mt. Gox (largest Bitcoin exchange at the time) declares bankruptcy (holdings of 850,000 bitcoins)

# HASH POINTERS

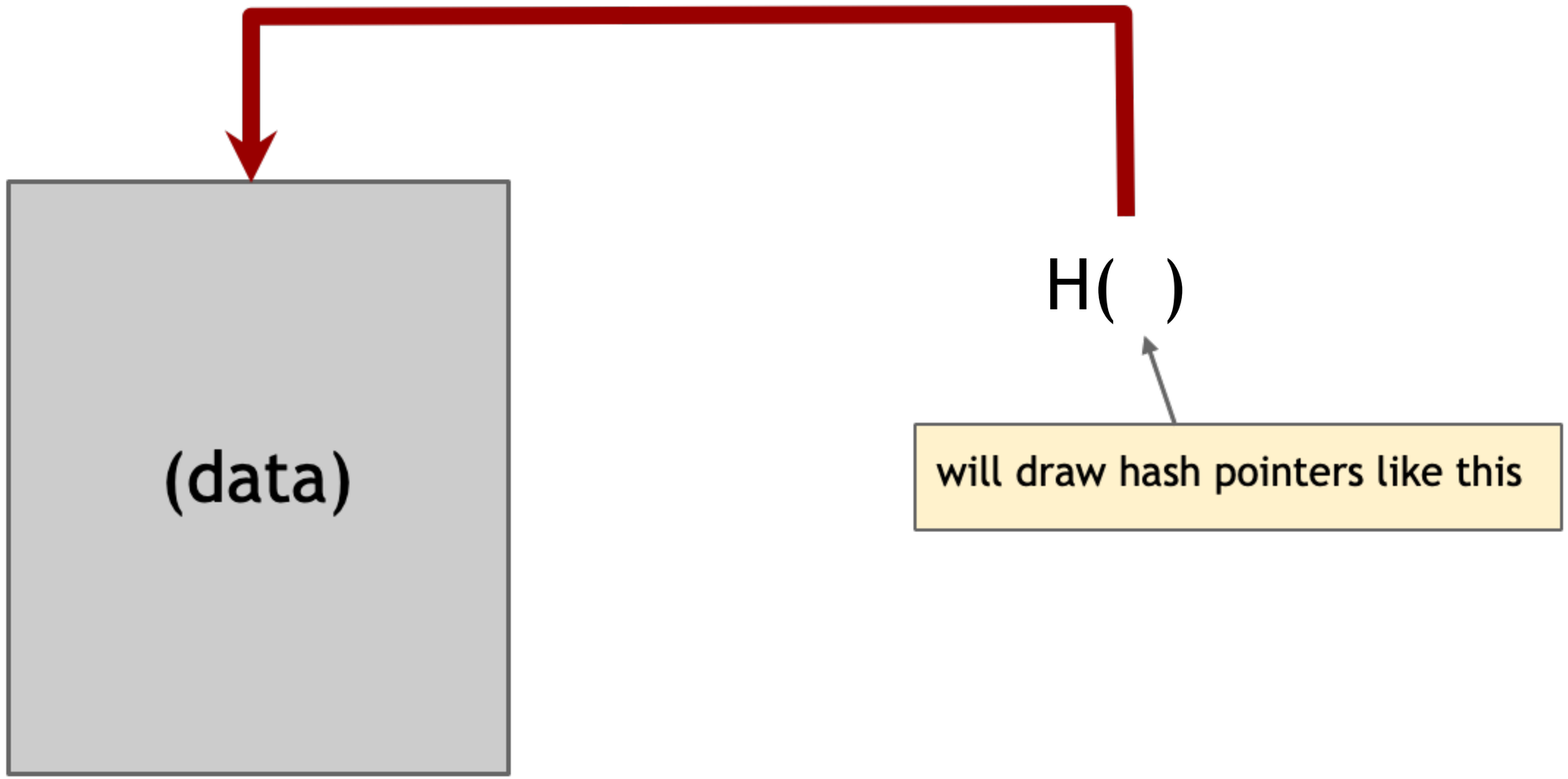
Given a collection of *data* at location *loc,* a **hash pointer** consists of

- a **pointer to** data location ***loc***, together with
- the **hash *H(data)*** of *data using a cryptographic hash function* $H : D \to R.$

Purpose of hash pointers:

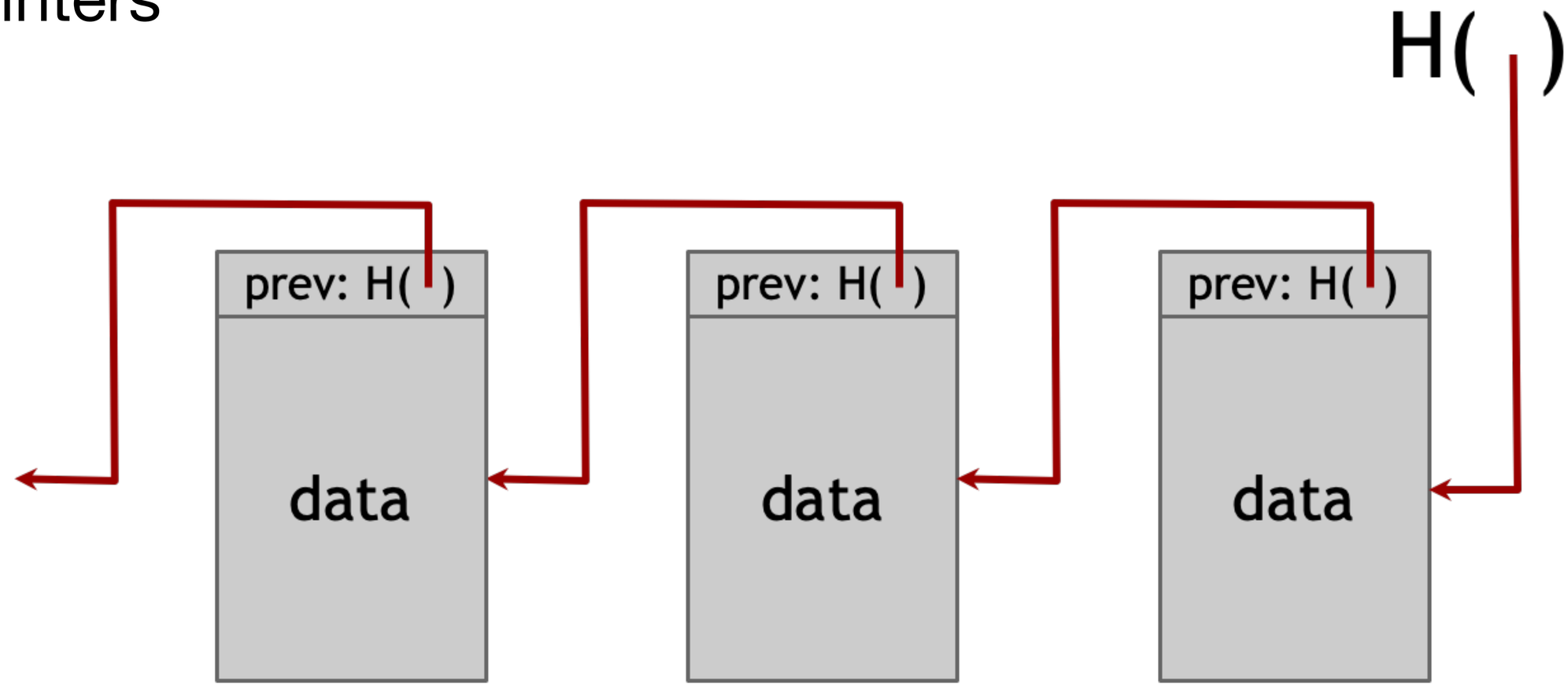
- Retrieve information of *data* if needed,
- Verify that information *data* has not changed.

# HASH POINTERS



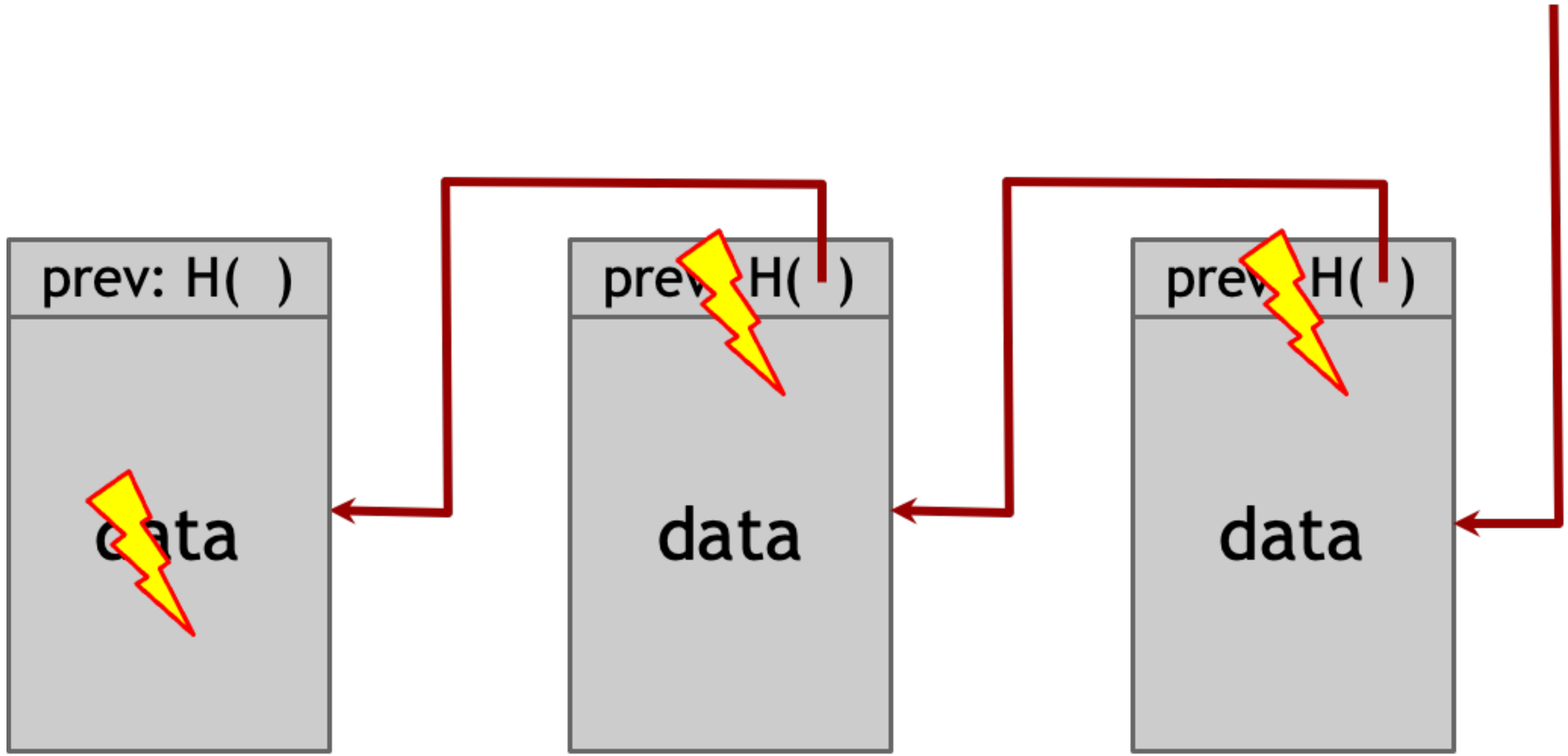
# BUILDING A "BLOCKCHAIN" OF DATA USING HASH POINTERS

Blockchain: Linked list of blocks of data linked through hash pointers



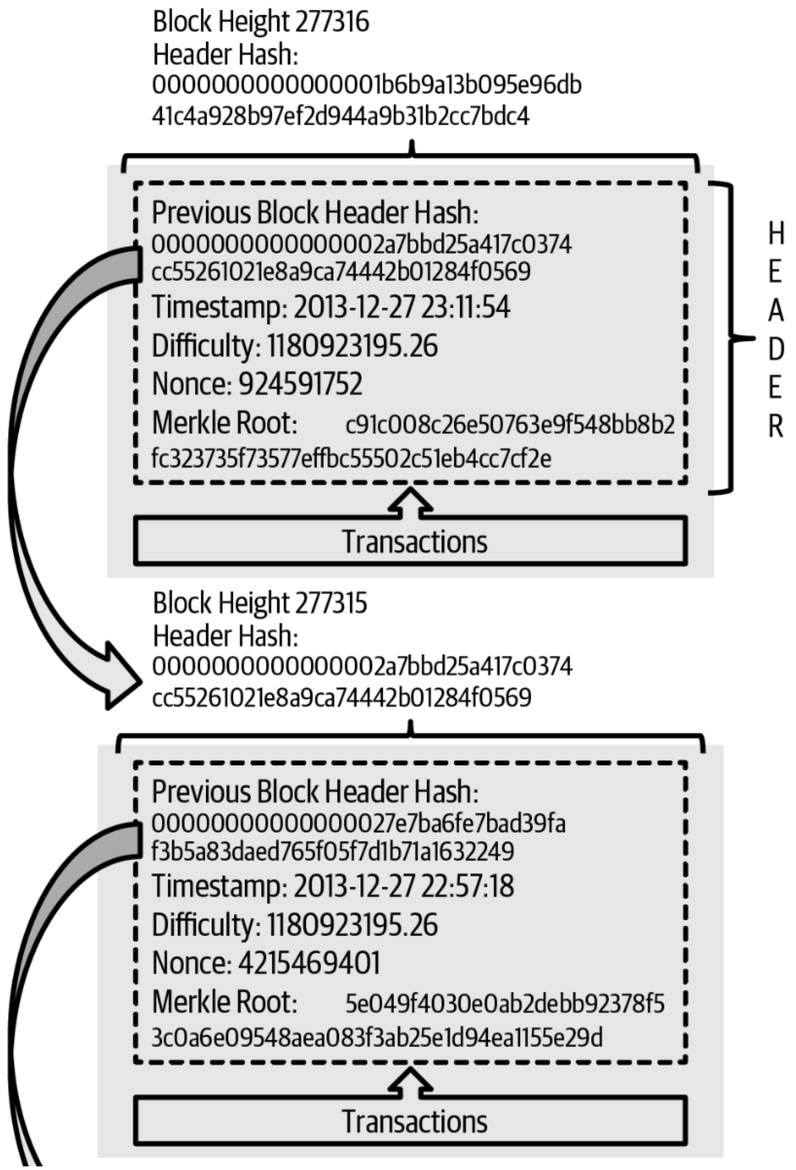

use case: tamper-evident log

# DETECTING TAMPERING IN A BLOCKCHAIN

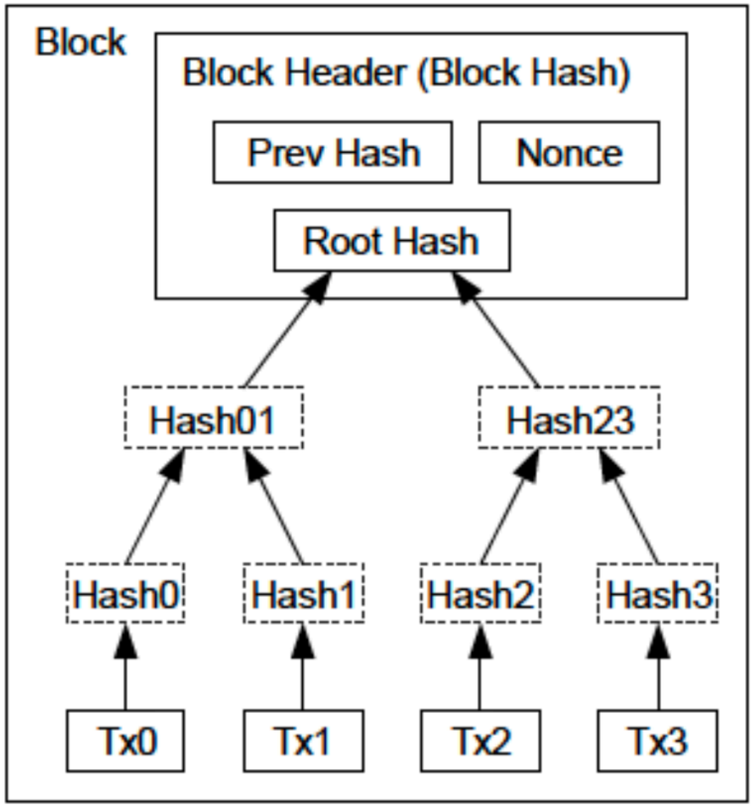



use case: tamper-evident log

# A BLOCK IN THE BITCOIN BLOCKCHAIN




Two Bitcoin blockchain blocks




Transactions Hashed in a Merkle Tree

Schematic structure of a Bitcoin blockchain block

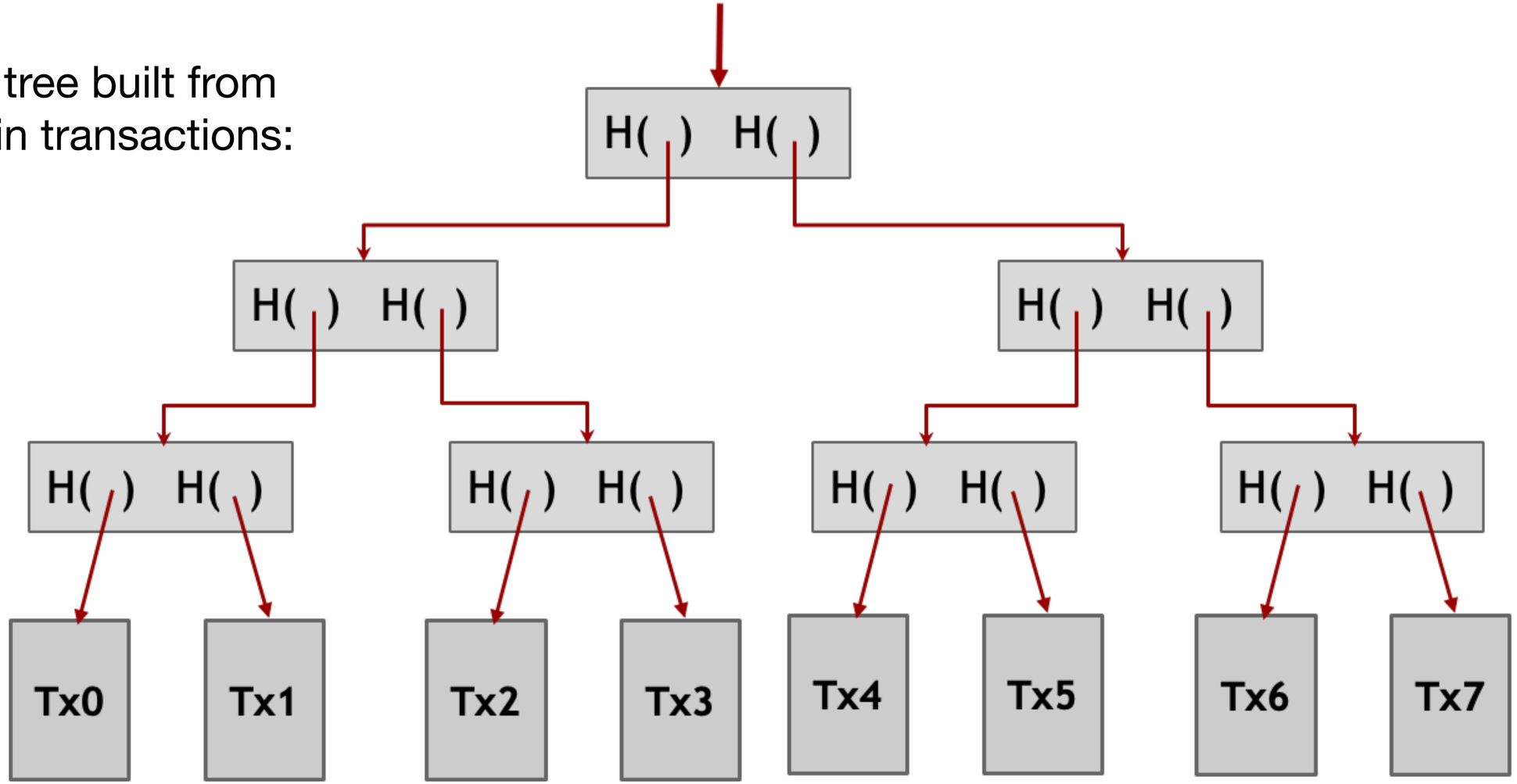
# BITCOIN BLOCK EXPLORER

- Website with visualizations of the Bitcoin blocks, memory pool (mempool) of pending transactions, etc.:

  https://mempool.space/

# A **Merkle tree** is a binary tree created through hash pointers

Merkle tree built from
8 bitcoin transactions:



# PURPOSE OF MERKLE TREES

Problem: Block might contain n transactions, n large

Merkle trees:

- Can verify membership of data in tree in O(log n) by providing “Merkle proof” of size O(log n)
- Sorted variant can prove non-membership of data in tree in O(log n)

# PROOF OF MEMBERSHIP IN MERKLE TREES

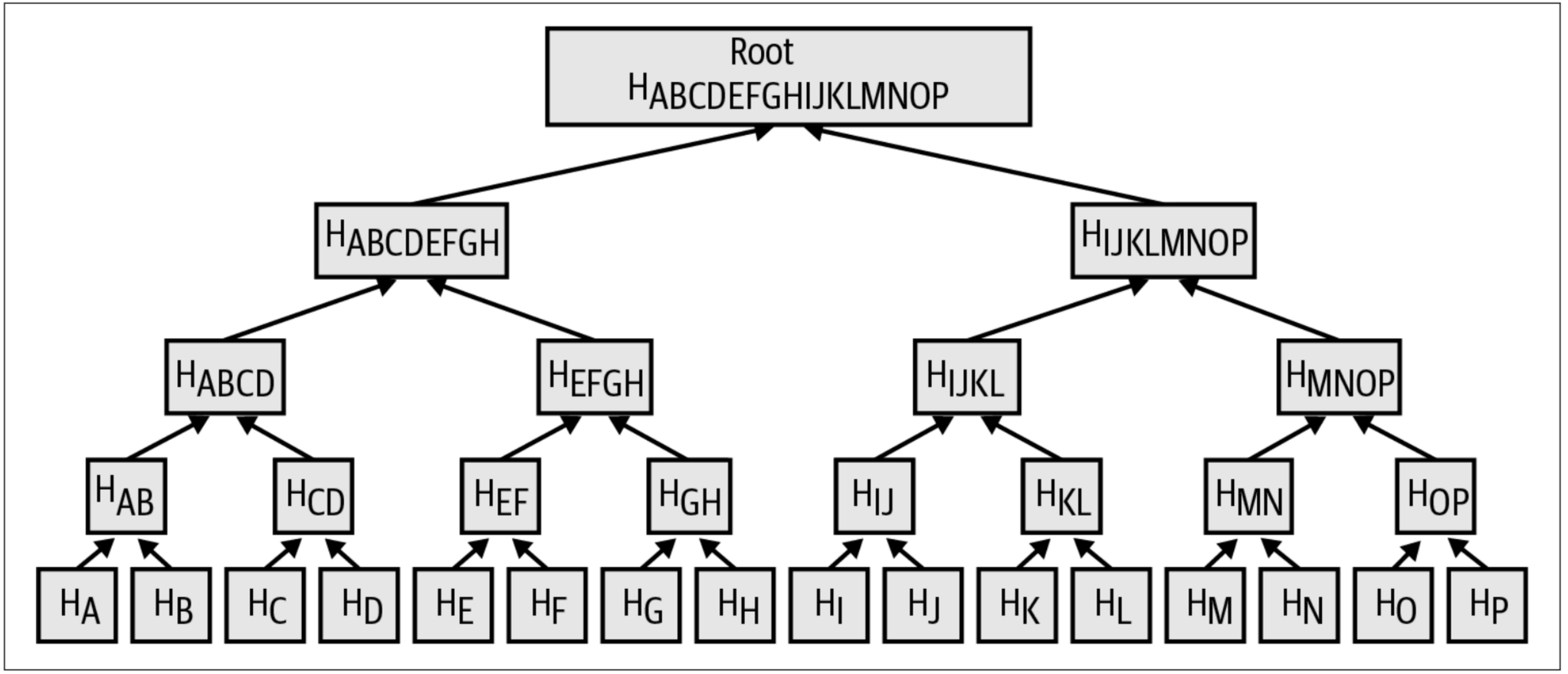



*Figure 11-5. A merkle tree summarizing many data elements.*

# PROOF OF MEMBERSHIP IN MERKLE TREES

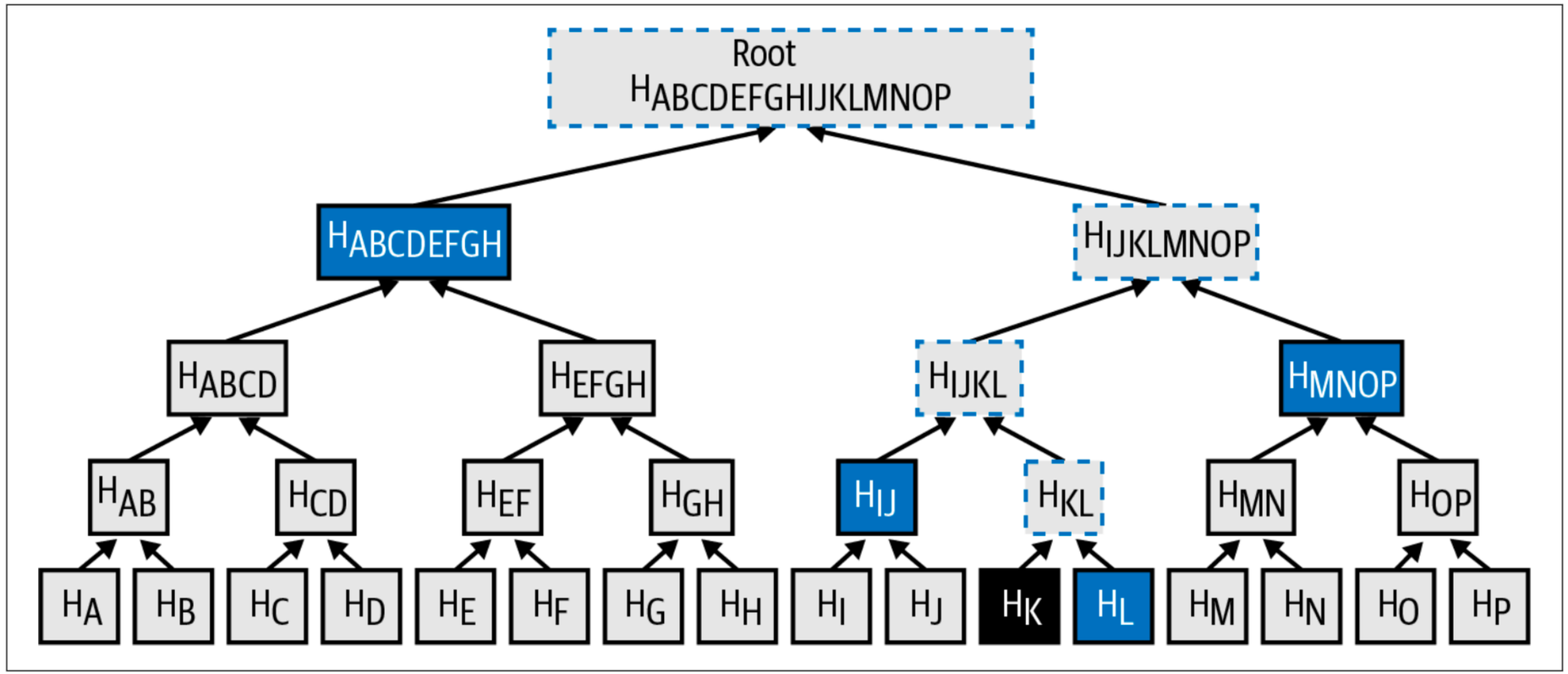



*Figure 11-6. A merkle path used to prove inclusion of a data element.*

# HOW DOES THE BITCOIN PROTOCOL WORK?

Questions:

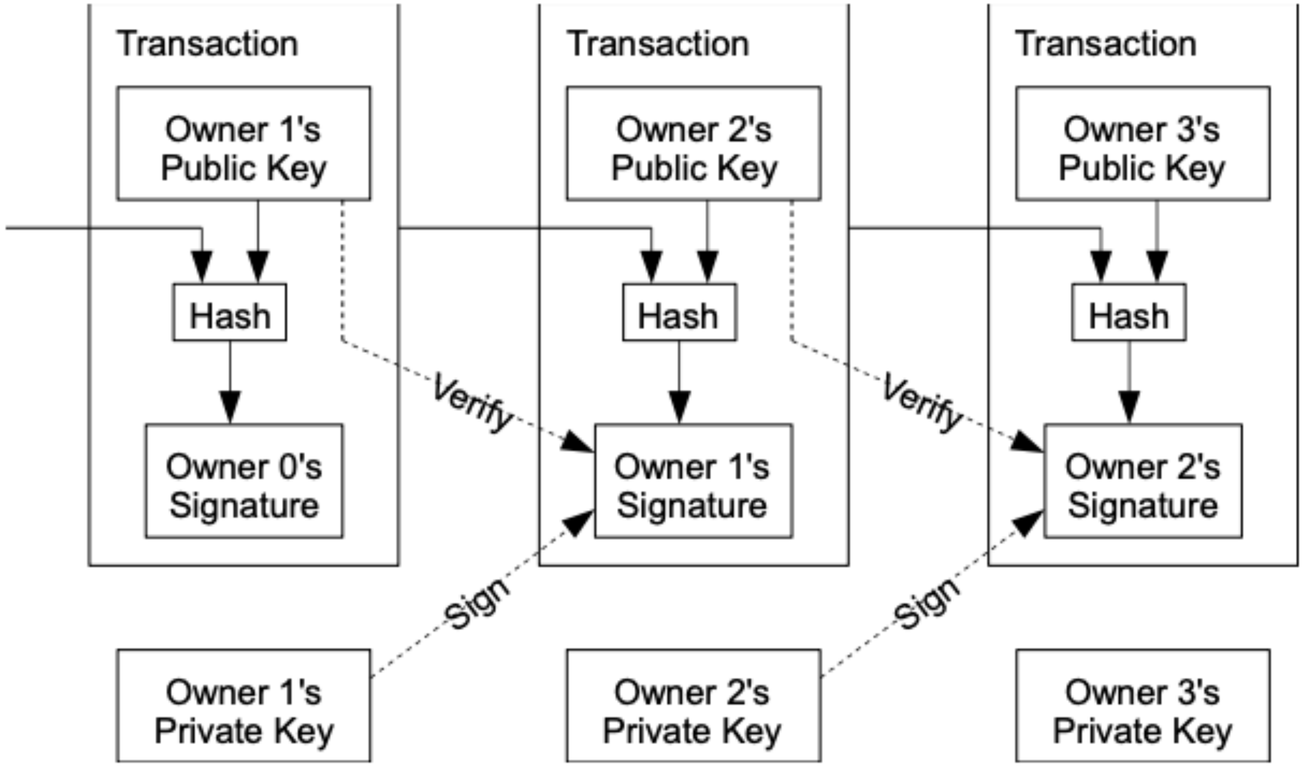
- How to add new blocks to blockchain? How do we know which blocks are the “right” ones?
  -> Mining, enforcement of rules by operators of Bitcoin node software
- What do the transaction look like, and what makes them valid?
- How do I “make” a valid transaction?
- How do I know which coins belong to “me” / or a specific indvidual?
- What is an identity in the Bitcoin protocol?
- Why is the whole system secure?

# HOW DOES THE BITCOIN PROTOCOL WORK?

Questions:

- How to add new blocks to blockchain? How do we know which blocks are the “right” ones?
  -> Mining, enforcement of rules by operators of Bitcoin node software
- What do the transaction look like, and what makes them valid?
- How do I “make” a valid transaction?
- How do I know which coins belong to “me” / or a specific indvidual?
- What is an identity in the Bitcoin protocol?
- Why is the whole system secure?

# A LOOK AT BITCOIN TRANSACTIONS




Relationship of successive Bitcoin transactions

# BITCOIN: HOW IT CONTINUED

- 2015-2017: Debate about whether/how to increase block size limit of 1MB to increase permitted transaction volume (“block size wars”)
- August 2017: **SegWit** (Segregated Witness) **update** “activated” by Bitcoin miners, implemented via **soft fork**:
  - Block size increased to 2MB, better scalability, new address format, removes transaction malleability

# THE BLOCKSIZE WAR

## “Big Blockers”

- Increase block size (from 1 MB to 8 MB) to allow for more transactions (supporting vision as “peer-to-peer cash”)
- Not everyone needs to run Bitcoin node
- Supported by **large Bitcoin businesses (exchanges/miners)**
- Implemented via **hard fork** on Aug 1, 2017 -> **Bitcoin Cash**

## “Small Blockers”

- Small block size (from 1 MB to up to 4 MB) increase as side effect of fixing some smaller issues
- Keeps incentive for decentralization, scaling through off-chain solutions
- Supported by **Bitcoin holders**
- Activated via **soft fork** on Aug 24, 2017 -> **Bitcoin (BTC)**


Book by Jonathan Bier, published in 2021.

# EXCHANGE RATE CHART - BITCOIN CASH (BCH) VS. BITCOIN (BTC)

# HARD AND SOFT FORKS

We distinguish:

- **Hard forks:**
  Change rules such that **new blocks / transaction types are not valid in the old rule set.**

-> Typically leads to chain split

- **Soft forks:**
  Change rules such that **new blocks / transaction types are still valid in the old rule set, but old transaction/block rules are not (necessarily) valid in the new rule set.**

-> Typically does not lead to chain split

# BITCOIN: HOW IT CONTINUED

- 2015-2017: Debate about whether/how to increase block size limit of 1MB to increase permitted transaction volume (“block size wars”)
- August 2017: **SegWit** (Segregated Witness) **upgrade** “activated” by Bitcoin miners, implemented via **soft fork**:
  - Block size increased to 2MB, better scalability, new address format, removes transaction malleability
- November 2021: **Taproot upgrade** implemented via soft fork:
  - New, more efficient signature scheme (Schnorr signatures)
  - Improves privacy of multi-signature transactions
  - Update to Bitcoin script language

- January 2024: U.S. Securities and Exchange Commission (SEC) **approves 11 spot Bitcoin Exchange Traded Funds** (ETFs): Increased acceptance of bitcoin in mainstream financial markets