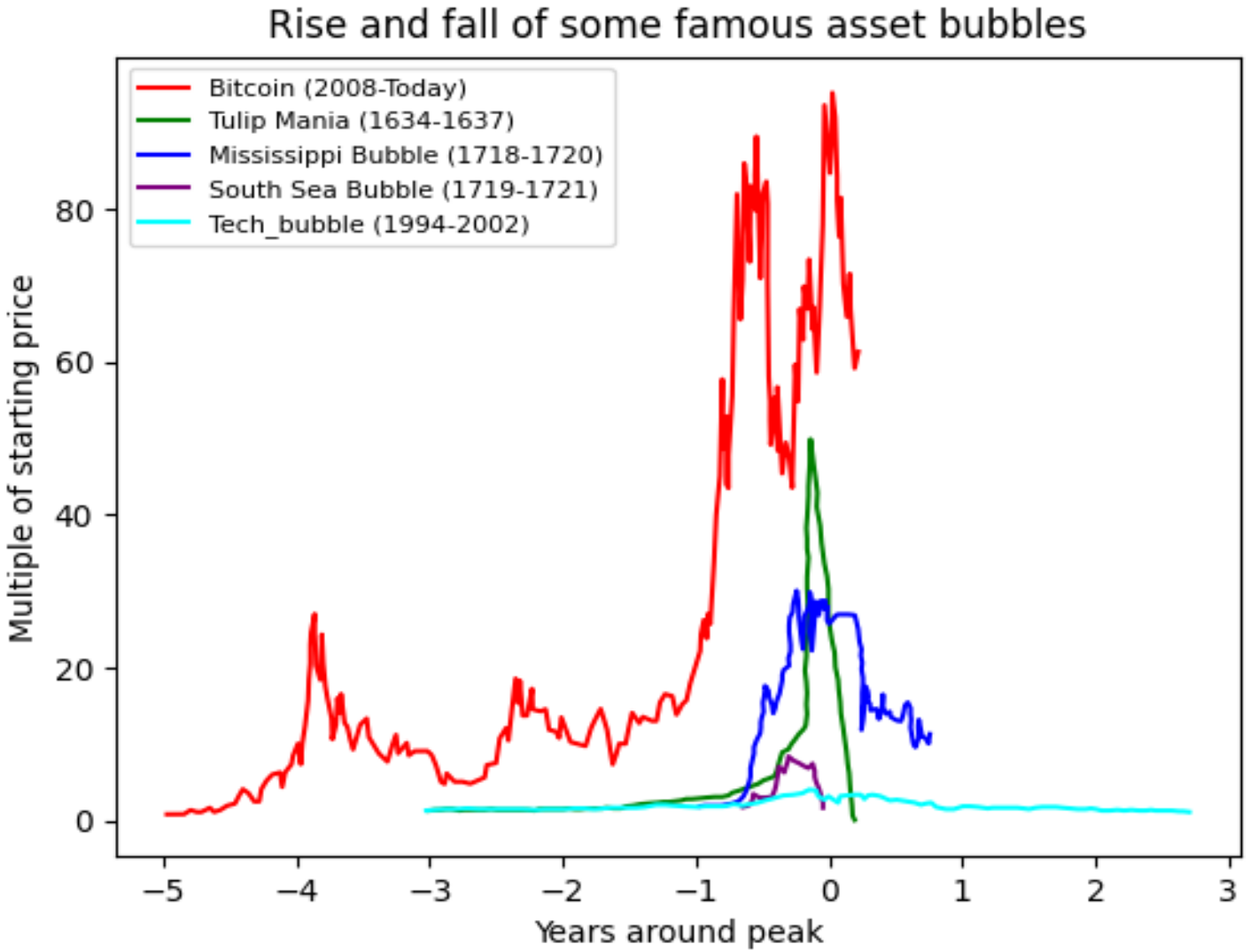
Bitcoin and Bubbles



Elements of DeFi

https://web3.princeton.edu/elements-of-defi/

Professor Pramod Viswanath

Princeton University

Lecture 2. Blockchain Primer

Blockchains are decentralized digital trust platforms

Ledger of ordered entries

(transactions, instructions of a computer program)

Blockchain characteristic: Open and Permissionless

1. This means anyone can participate

2. Two types of participation

   - User – make transactions, token tx/rx, write/use dApps
     This class (ECE/COS 473)

   - Miner or Validator – maintain the underlying common ledger
     Separate class (ECE/COS 470)

Blockchain characteristic: Open and Permissionless

1. Open source: code is public
2. Open state: memory (ledger) is shared and publicly viewable
3. Open entry: anyone can “call” any code
4. Open exit: automated execution
5. Open participation: permissionless

Technically challenging to maintain trust

Cryptography

Provides basic tools to address both design goals

Challenge: “The trouble is, the other side can do magic too, Prime Minister.”

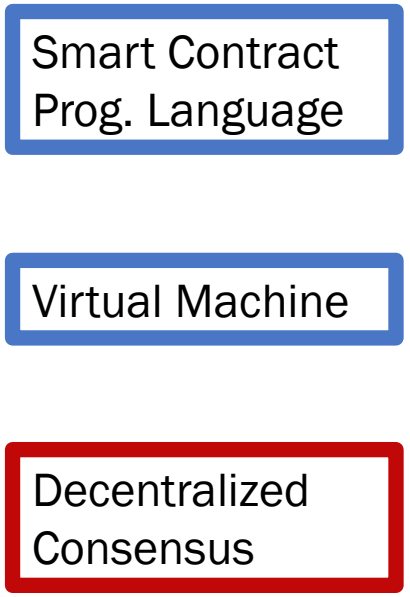
Blockchain: Distributed Computing Platform

- Platforms are applications built on networked computers
- Basic building block: **Decentralized Computer**

1. Multiple untrusted computers interacting with one another, forming **consensus** on an ordered list of instructions
2. A **virtual machine** interprets the instruction set
3. A programming language and a corresponding compiler provide a forum for **decentralized applications (dApps)**

Technical Components

- Decentralized Computer
  - Cryptographic data structures
  - Disk I/O and Database management
  - Memory management
  - Operating systems
  - Peer to peer networking
  - Consensus and distributed algorithms
- Virtual Machine
  - Reduced instruction set, incentives
  - General purpose programming language



Nearly all aspects of Computer Science

Blockchain Participation

- Identity
  - Cryptographic digital signatures
- Participation in forming and verifying ledger of ordered entries
  - “Resource” or “Stake” or “Collateral” to prove seriousness
  - Mining
  - Verification

Digital Signatures

Key generation

(secretkey, publickey) =
Generatekeys(keysize)

Randomized function

Signature

Sig = *sign*(secretkey, message)

Verification

verify(publickey, Sig, message)

Unforgeable Signatures

Unforgeable

Computationally hard to generate a verifiable signature without knowing the secret key

ECDSA

Elliptic Curve Digital Signature Algorithms

Cryptographicaly secure against an adaptive adversary

Signatures in Practice

Elliptic Curve Digital Signature Algorithm (ECDSA)
Standard part of crypto libraries

Public key: 512 bits

Secret key: 256 bits

Message: 256 bits
Note: can sign hash of message

Signature: 512 bits

Decentralized Identity Management

Public keys are your identity
    address in Bitcoin terminology

Can create multiple identities
    (publickey, secretkey) pairs
    publish publickey
    sign using secretkey

Can create oneself
    verifiable by others

State Management

- **State**: which address owns how much tokens

- **UTXO model**
  - Unspent transaction output
  - Used in Bitcoin
  - Especially suited for managing plain transactions

- **Account model**
  - (address, amount)
  - Used in Ethereum and most blockchains
  - Especially suited for managing *instructions* (**dApps)**

Consensus

- **Ledger**: ordered list of *instructions*

- **Agreement**
  - Participants agree on the ledger
  - Once agreed upon, there is no reneging

- **Liveness**
  - Ledger should be allowed to grow (new transactions)
  - Any valid transactions should eventually enter the ledger (no censoring)

- **Proof of Participation**
  - Participants need to have some “stake” (i.e., collateral)
  - Proof of Work, Stake, Space

Distributed Consensus

Question: Who maintains the ledger of transactions? How is it built?

Distributed Consensus

Interactive Protocol

Allows distributed non-trusting nodes to come to agreement

Traditional area of computer science (Byzantine Fault Tolerance)

Bitcoin's consensus protocol is vastly different

decentralized identity (permissionless setting)

less pessimistic network assumptions

Agreement on a Block

Time is organized into **slots**

Leader election

- **Oracle selects one of the nodes** (public identities)
  - random, idealized functionality
  - everyone can verify the unique *winner*

The selected node (leader) is the proposer in that slot

- constitutes a block with transactions
- validates transactions
- signs the block

Blockchain: Chaining blocks together

Blockchain as a data structure

hash pointer based linked list

Hash functions

Hash pointers

Hash Function

1. Arbitrary sized input
2. Fixed size, uniform output
3. Simple deterministic function
4. Collision resistant

Example: Division hashing $y = x \mod 2^{256}$

Cryptographic Hash Functions

Extra Properties:

1. Adversarial collision resistance

2. One way function

Hash Pointer

Hash of the information acts as pointer to location of information

Regular pointer: retrieve information

Hash pointer: retrieve information and verify the information has not changed

Regular pointers can be used to build data structures: linked lists, binary trees.

Hash pointers can also be used to build related data structures. Crucially useful for blockchains. In fact, **blockchain itself is a hash pointer-based data structure.**

Blockchain: a linked list via hash pointers

Block: Header + Data

Header: Pointer to previous block
= hash of the previous block

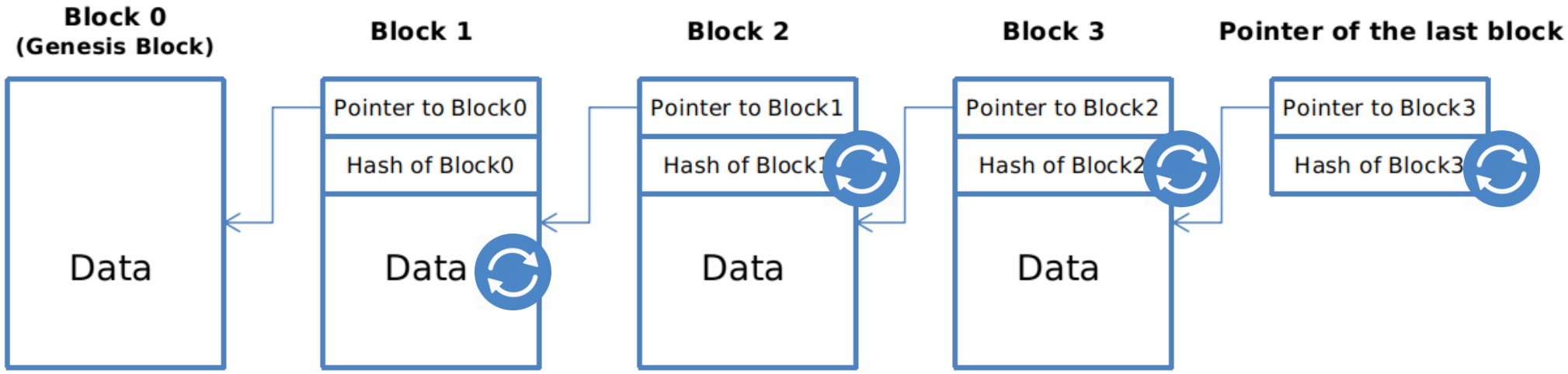
Data: information specific to the block

Application: tamper evident information log

Head of the chain being known is enough to find tamper evidence in any internal block

Hence the phrase: **block chain**
blockchain

Blockchain: a linked list via hash pointers



Allows the creation of a tamper-evident information log

How about searching for specific data elements?

Proof of Work

Practical method to simulate the Oracle

Mining

cryptographic hash function creates computational puzzle

Hash(nonce, block) < Threshold

nonce is the proof of work

include nonce inside the block

Threshold

chosen such that a block is mined successfully on average once in 10 minutes

a successfully mined block will be broadcast to all nodes in the network

Properties of Proof of Work Mining

1. Random miner selected at each time

2. Independent randomness across time and across miners

3. Probability of successful mining proportional to fraction of total hash power

4. Sybil resistance

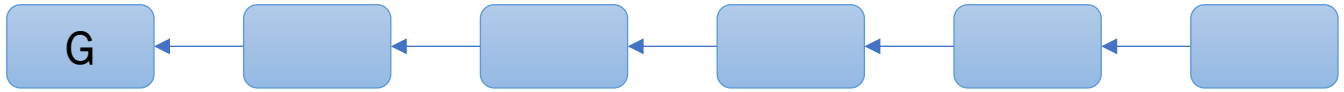
5. Spam resistance

6. Tamper proof – even by the proposer!

Longest Chain Protocol

Where should the mined block hash-point to?

The latest block



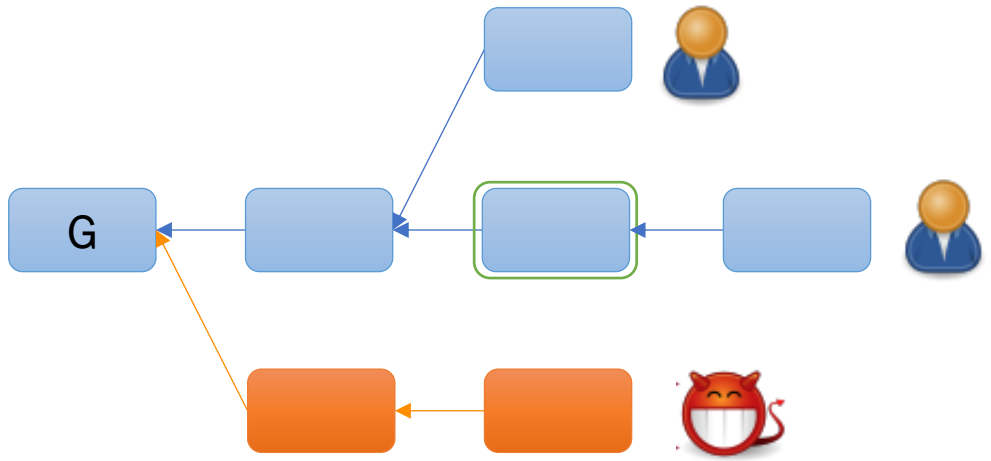
Longest Chain Protocol

Where should the mined block hash-point to?

However, blockchain may have **forks**

because of network delays

because of adversarial action



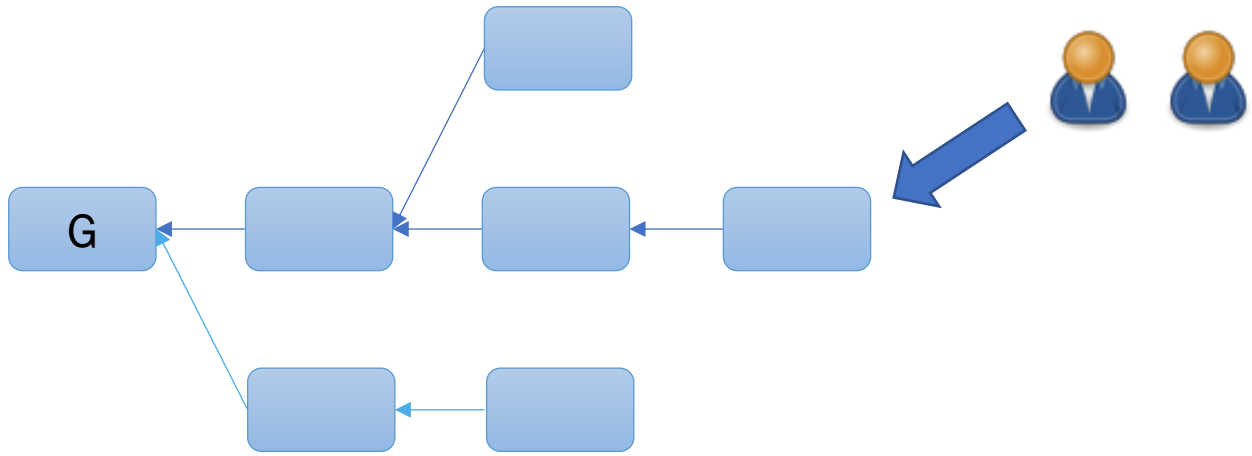
Longest Chain Protocol

Where should the mined block hash-point to?

Blockchain may have **forks**

because of network delays

because of adversarial action



Longest chain protocol

attach the block to the leaf of the longest chain in the block tree

Security Analysis: Private Attack

Adversary can point its block to an older part of the chain

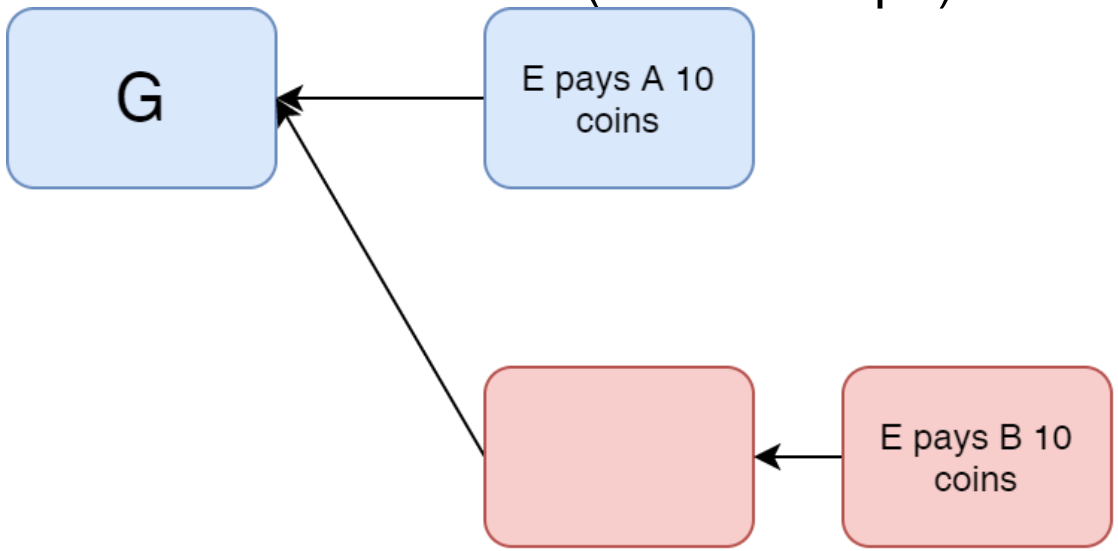
Duplicate transaction inserted

Plausible Deniability

network latency

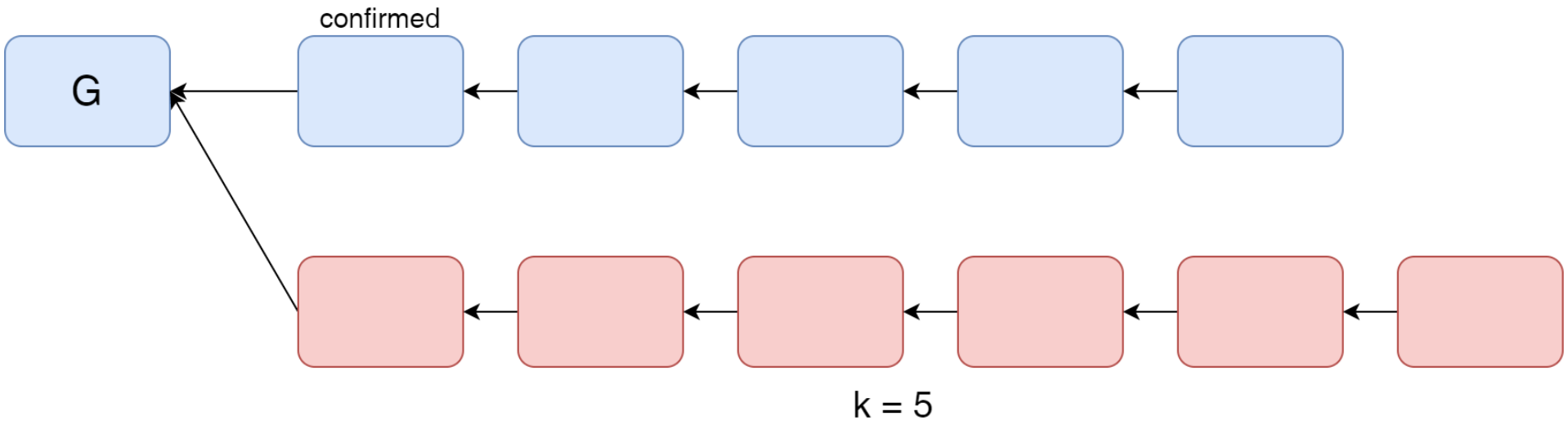
an offline user will not know which block came earlier

blocks have no wall clock reference (time stamps).

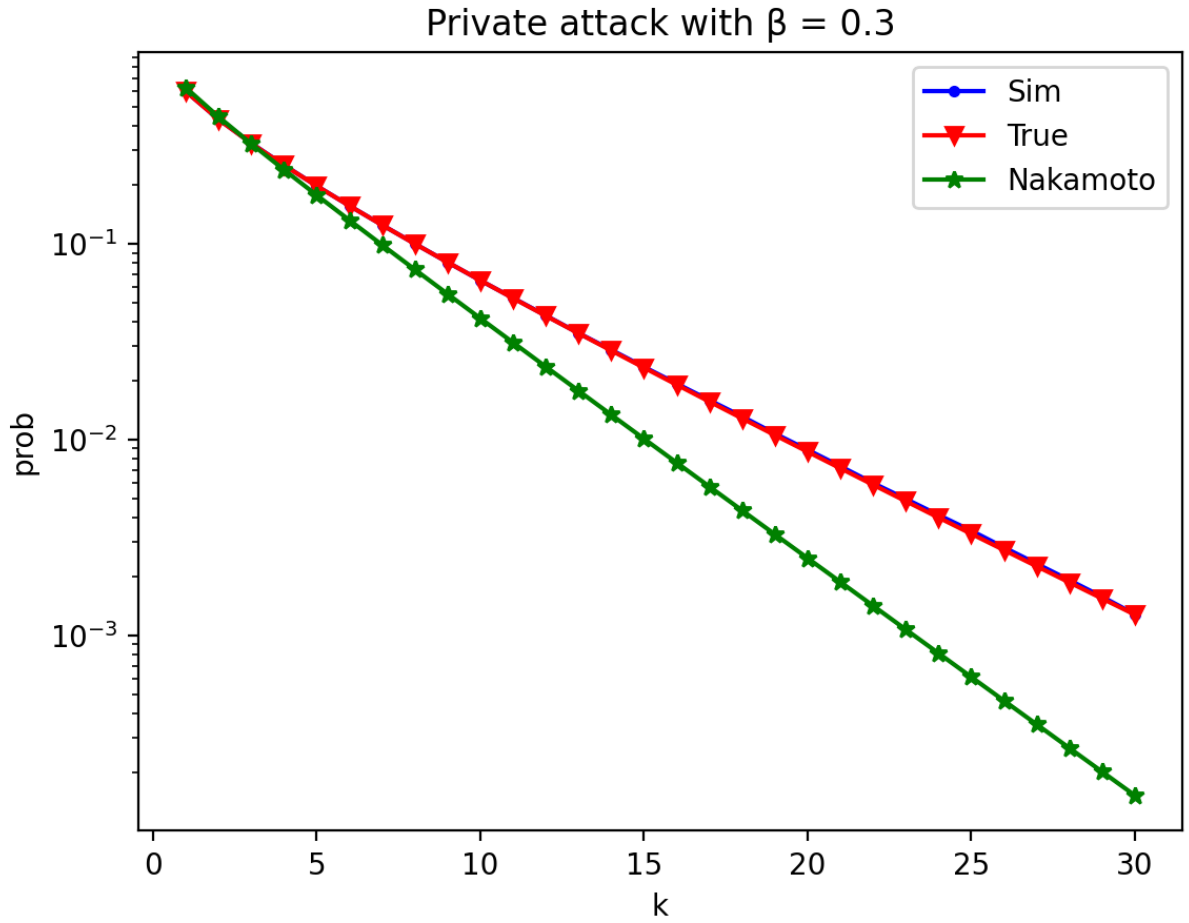


Security Analysis: k-deep Confirmation Rule

- A block is confirmed if it is buried k-deep in the longest chain
- An attacker would need more than k blocks to double spend



Security vs Latency with Private Attack



Pros and Cons of the Longest Chain Protocol

- Liveness
  - Even a single honest miner with a small hash power can extend the longest chain
- Safety
  - Guaranteed when hash power of honest nodes is more than 50%
  - But with 2 caveats
    - Probabilistic guarantee
    - Network must be synchronous

Byzantine Fault Tolerant (BFT) Protocols

- Deterministic safety even under asynchronous network

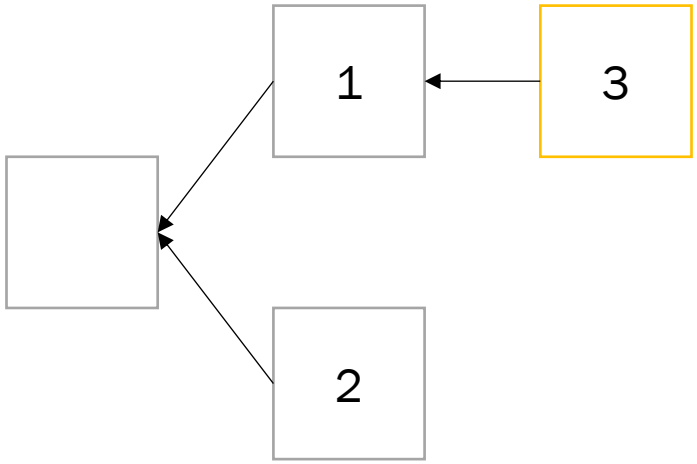
- Two closely related protocols:
  - Streamlet
  - HotStuff

BFT Protocol Setting

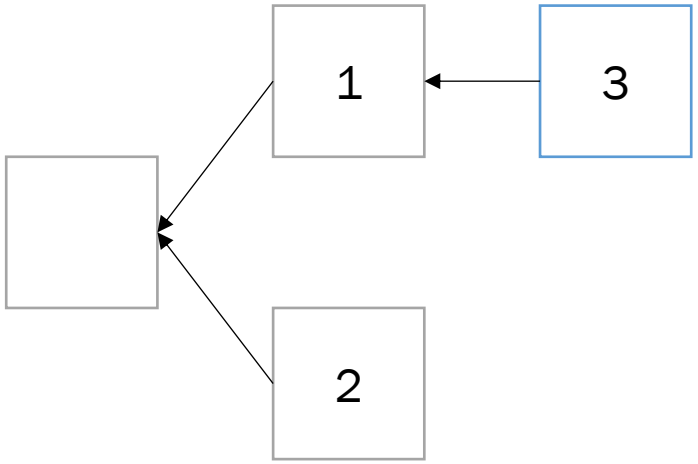
- Number of Participants is fixed
- Identities known (signatures) to all nodes
- In other words, “permissioned”

BFT Steps (Round-by-Round)

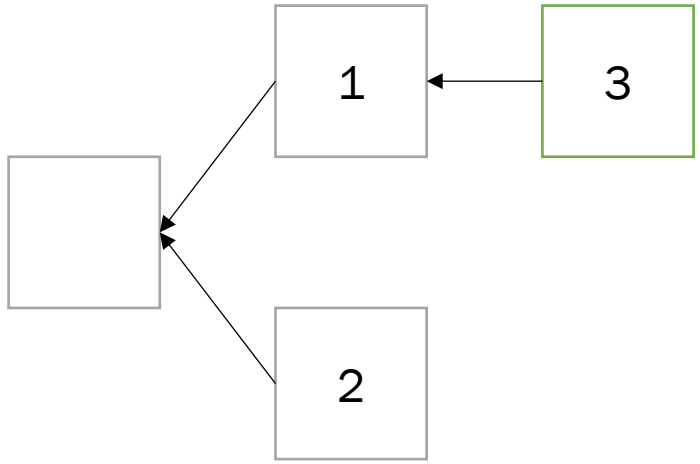
- In each round:



1. Propose a new block



2. Vote



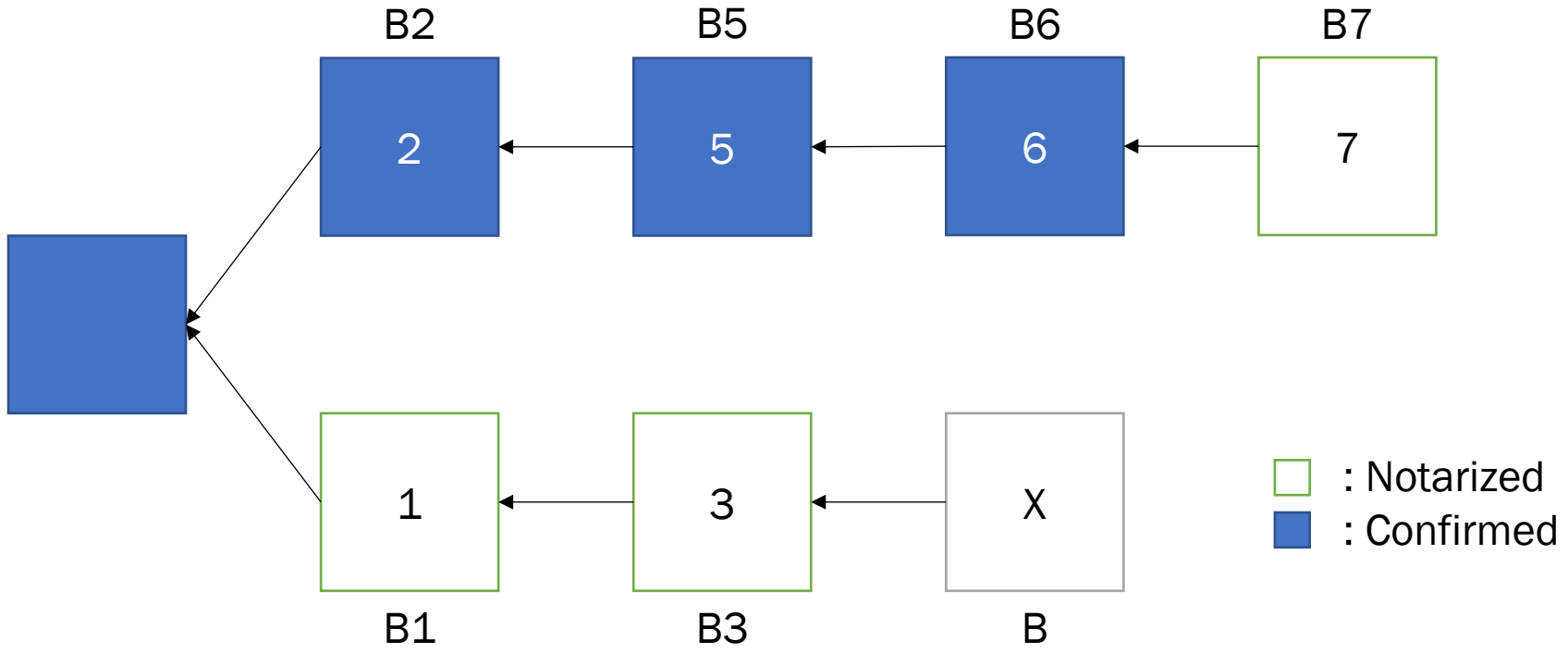
3. Notarized

Streamlet

- Proceeds in lock-step rounds, each of which takes twice the communication delay
  - One leader elected every round
  - Leader will collect pending Tx and put a block together and propose
  - Whenever a block is proposed
    - Checks if signed by leader with rights to propose in round r
    - Vote on a proposed block if the block extends the longest notarized chain (a block is notarized if it receives at least 2N/3 votes)
- All nodes re-broadcast all messages they hear of
- A node does not vote for conflicting blocks

Streamlet Confirmation Rule

- Correct rule: On seeing three adjacent blocks in a notarized blockchain with consecutive round numbers, a player can confirm the second of the three blocks, and its entire prefix chain.



Blockchain and Consensus Protocols

- **Ethereum**
  - Originally same as Bitcoin (Proof of Work, Longest Chain Rule)
  - Merge (9/15/22): Proof of Stake, Complicated longest chain rule
- **Avalanche, Cardano, Cosmos, Solana**
  - Proof of Stake
  - Different consensus protocols
- **Scaling solutions**
  - Polygon, Arbitrum
  - Outsource execution, storage but retrain trust in original blockchain
- **Interoperability**
  - Blockchains largely in individual silos
  - Bridges – active area of research and development

ECE/COS 470: Principles of Blockchains

- This course presents the **design space of blockchains**
  - **Principles** of good blockchain design choices
  - **Full-stack view**

- Pre-requisite: maturity with nearly all aspects of computer science

- Concretely: basic background in algorithms, probability, systems programming