PennState

CSE 543: Computer Security
Module: Bitcoin

Asst. Prof. Syed Rafiul Hussain

Department of Computer Science and Engineering

Intro and Basic Concepts

- Bitcoin is the technology that started it all
  - Bitcoin is a cryptocurrency

- Blockchain is the technology underlying Bitcoin
  - Enables distributed consensus

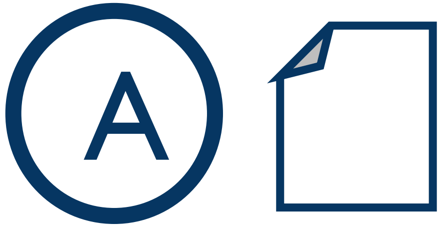
- Community terminology
  - Crypto, cryptocurrency – Bitcoin, Ethereum, more technical
  - Private blockchains, permissioned ledgers, or just "blockchain"
  - Distributed tech or decentralized tech – umbrella term

Satoshi Nakamoto’s Innovation

- Bitcoin was created by Satoshi Nakamoto in 2009
  - Decentralized, trustless systems for transactions
    - A low cost fianancial system that only requires an internet connection
  - Nakamoto solved the Double Spending problem
    - Prevent someone from spending the same asset twice
    - Solution? The Blockchain + Proof-of-Work

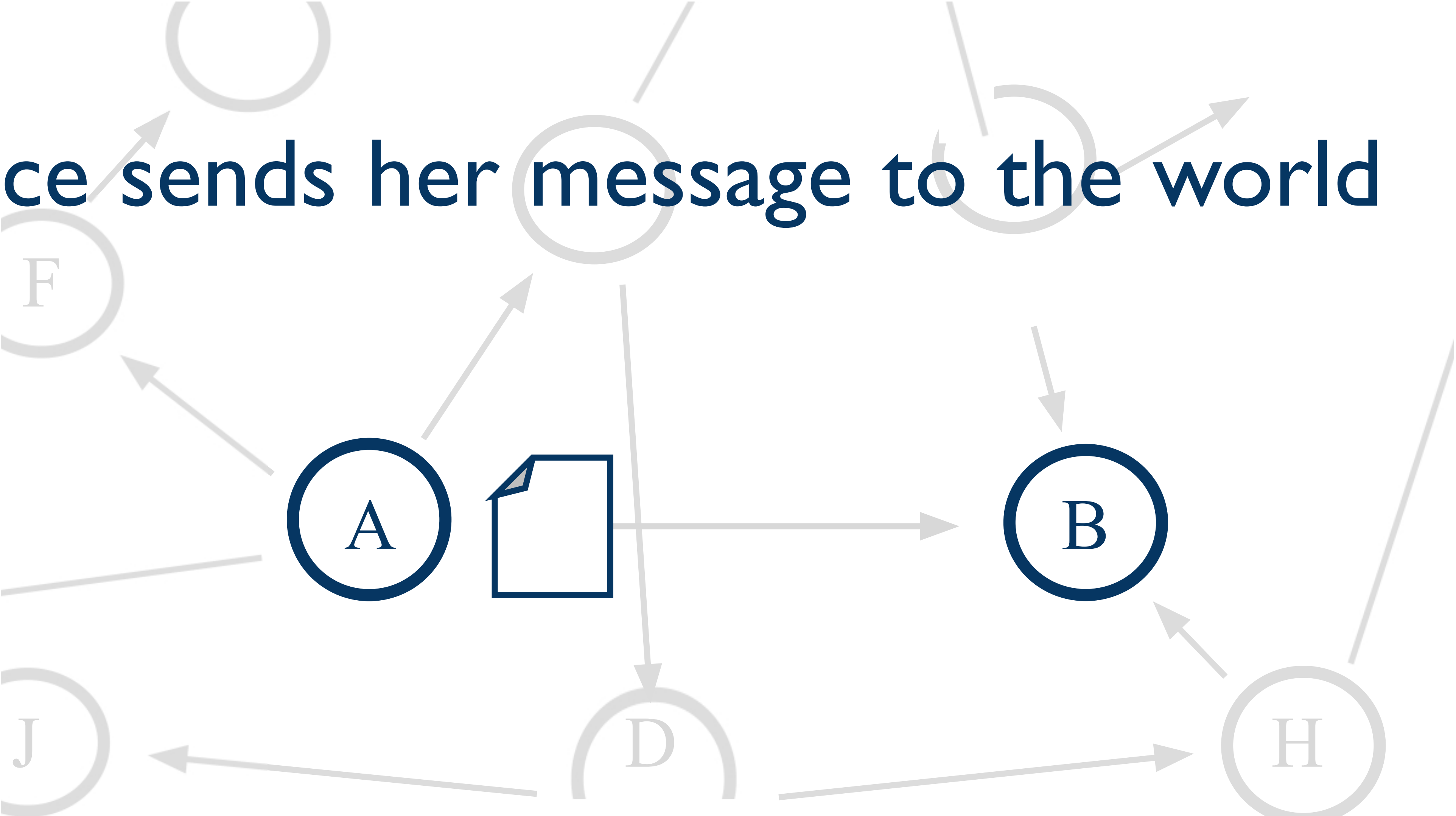
Dorian Satoshi Nakamoto
(not actually Satoshi Nakamoto)

Alice writes and signs a message describing her transaction

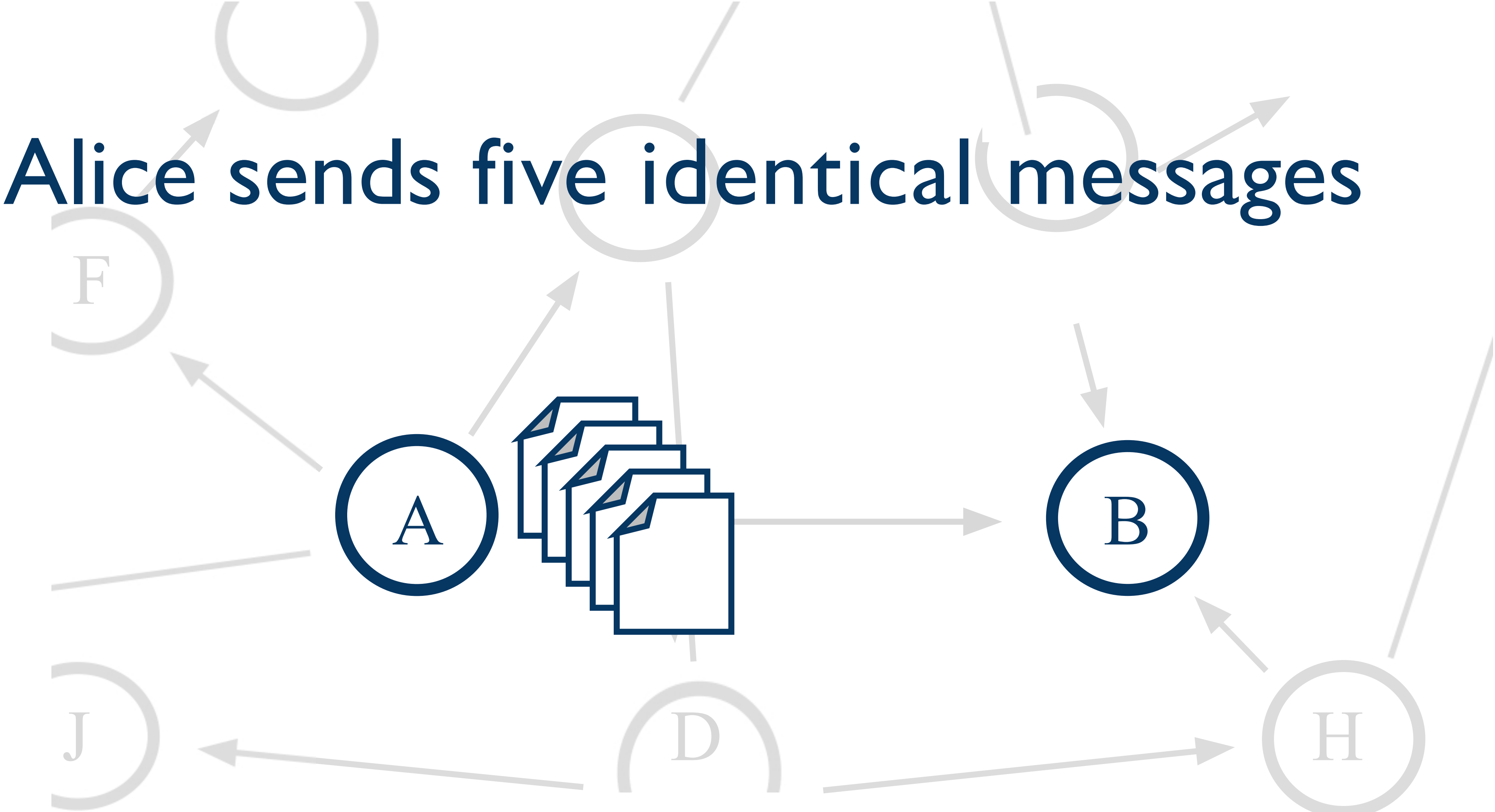


"I, Alice, am giving Bob one bitcoin."

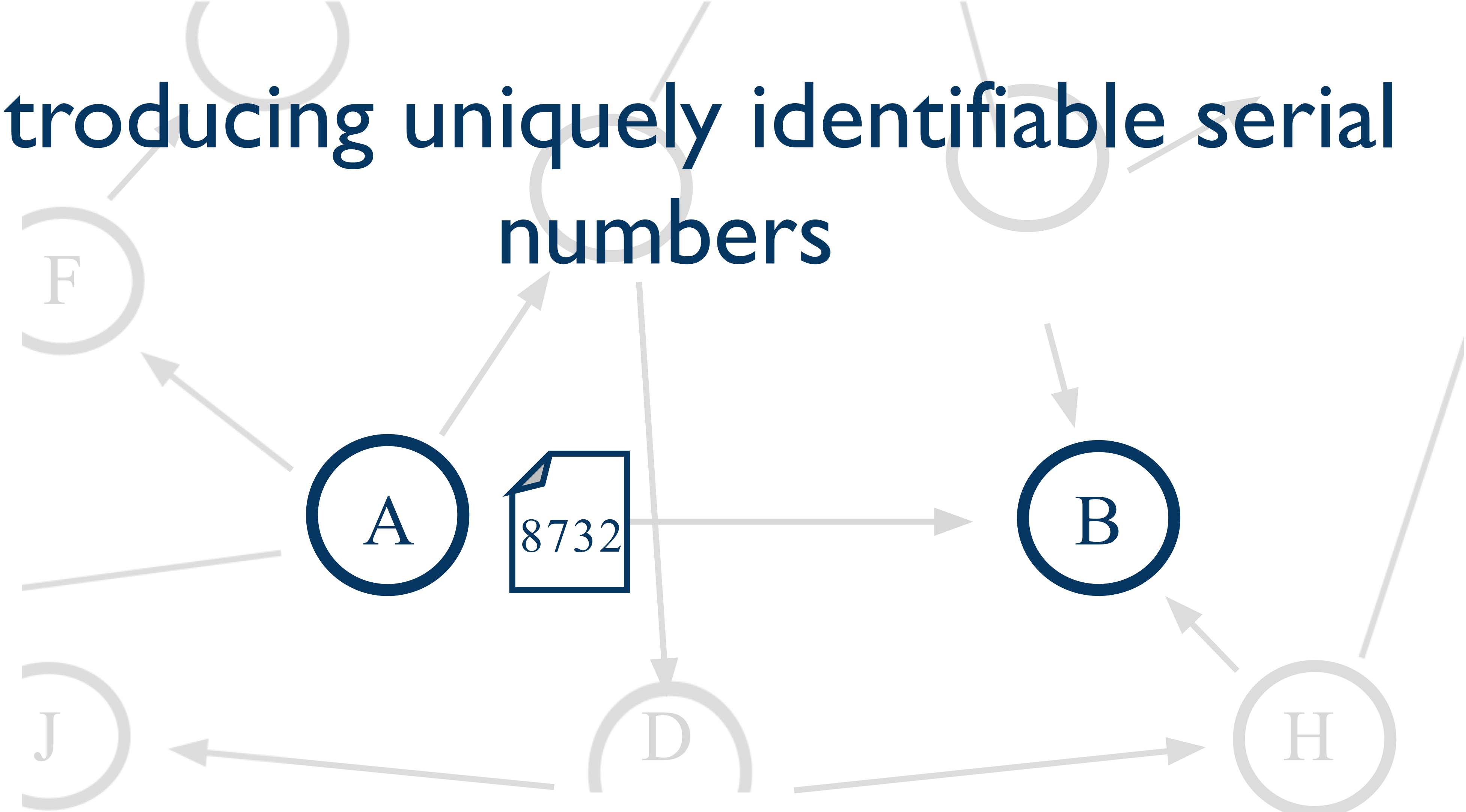
Alice sends her message to the world



Alice sends five identical messages

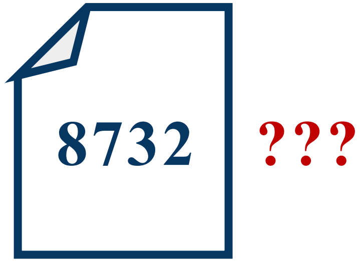


Introducing uniquely identifiable serial numbers

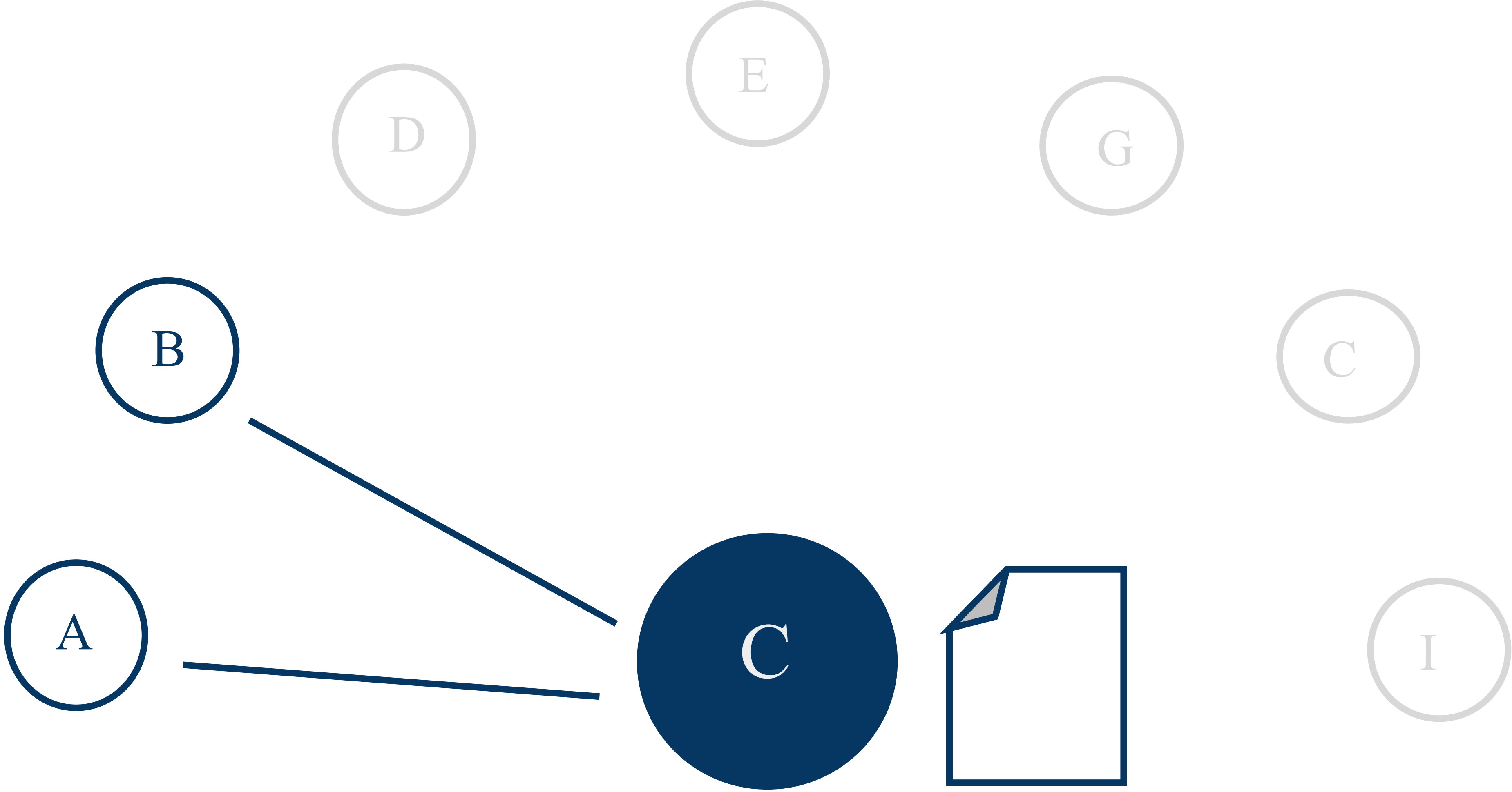


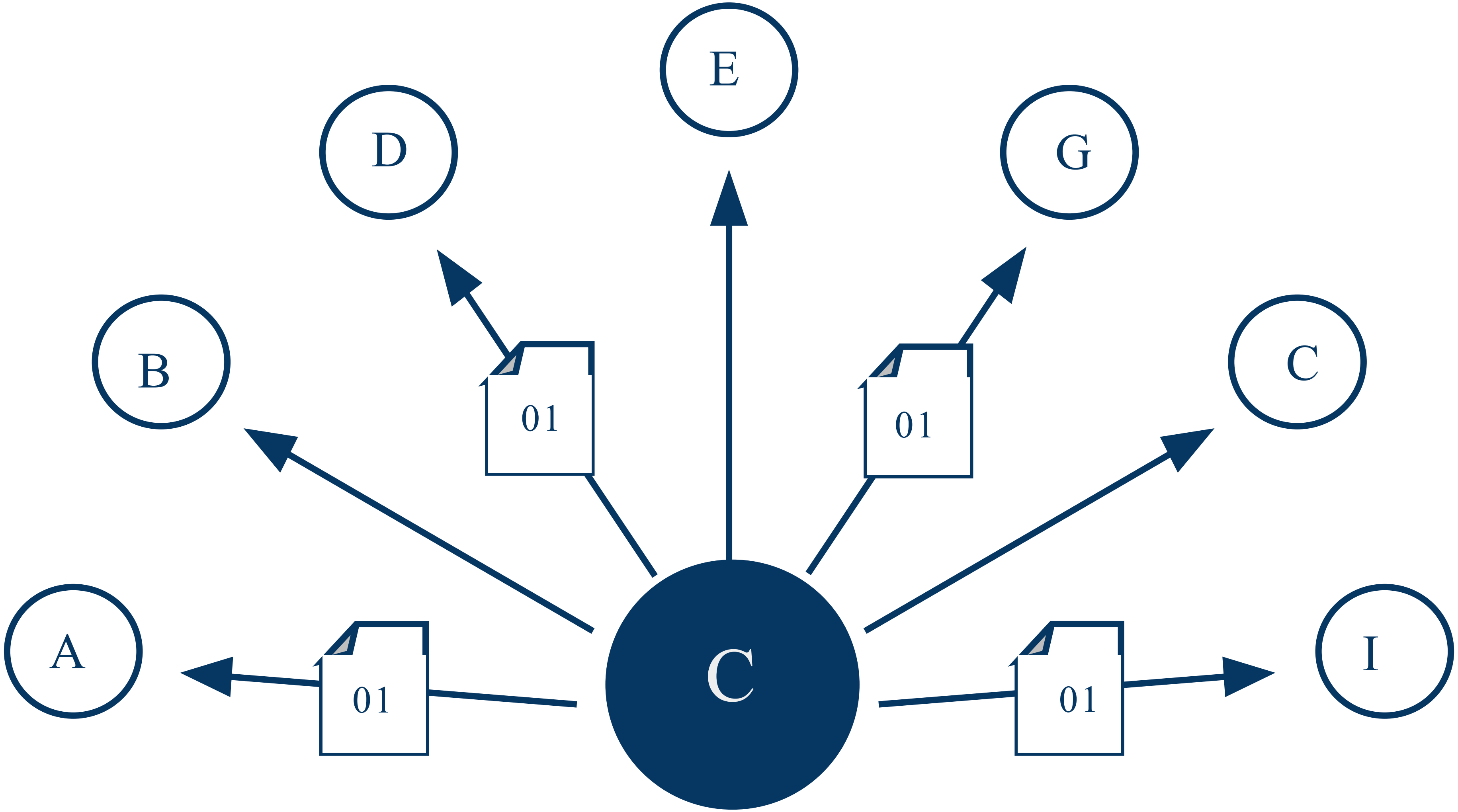
V2

Where do serial numbers come from?



A central bank manages transactions and balances

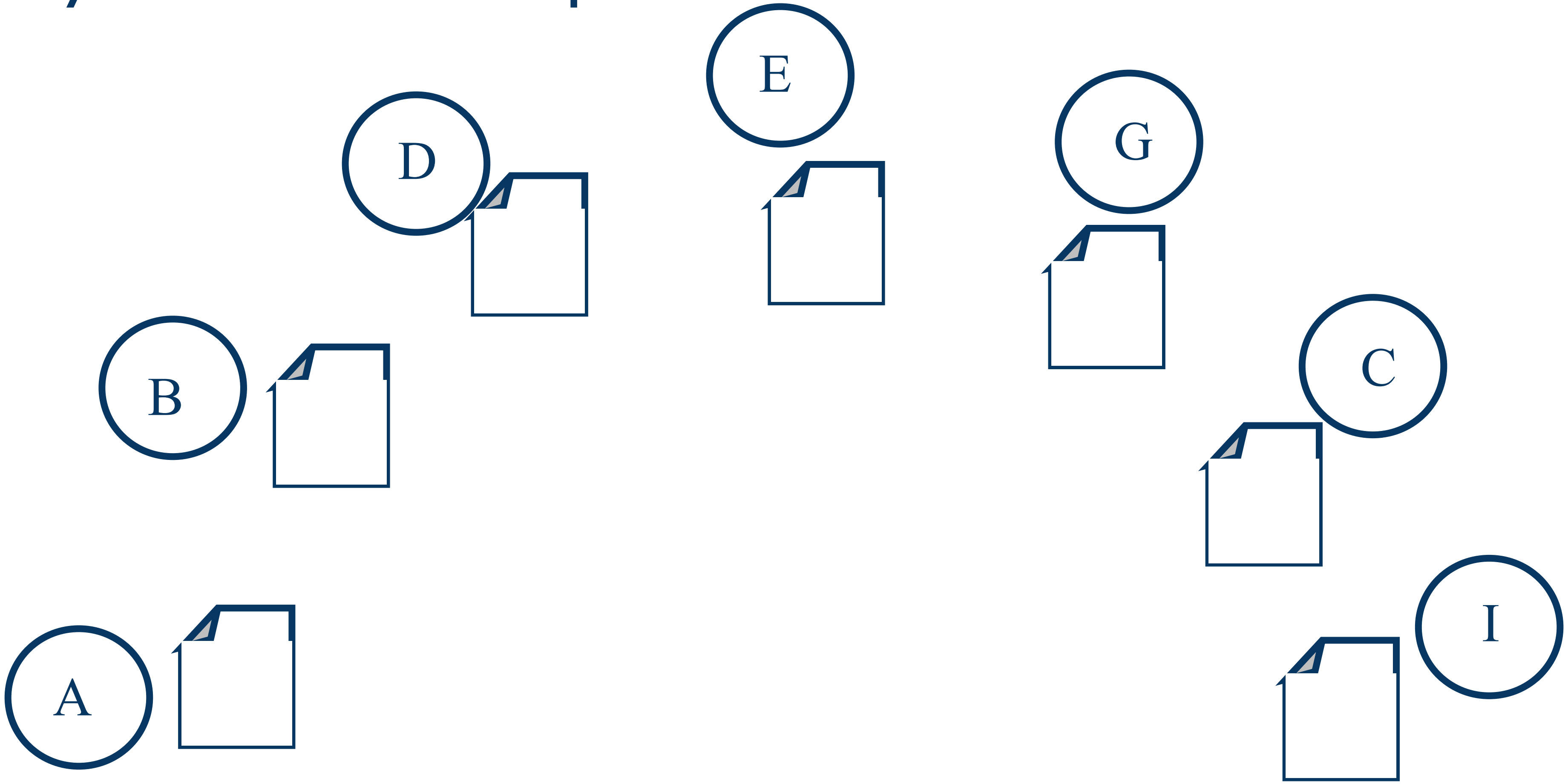



E
D
G
B
01
01
C
A
01
C
01
I

V3

Making everyone the bank.

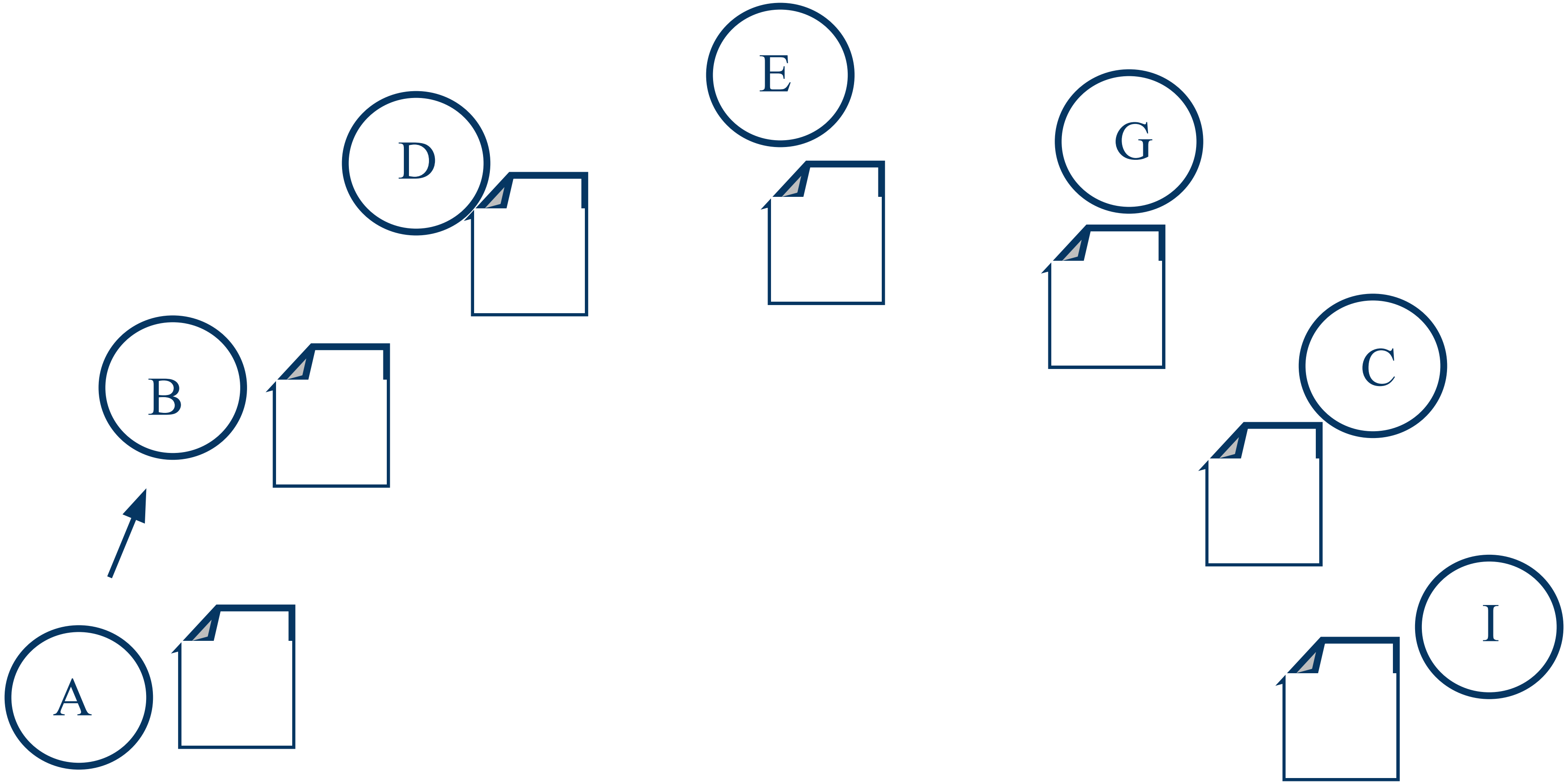
Everyone has a complete record of transactions



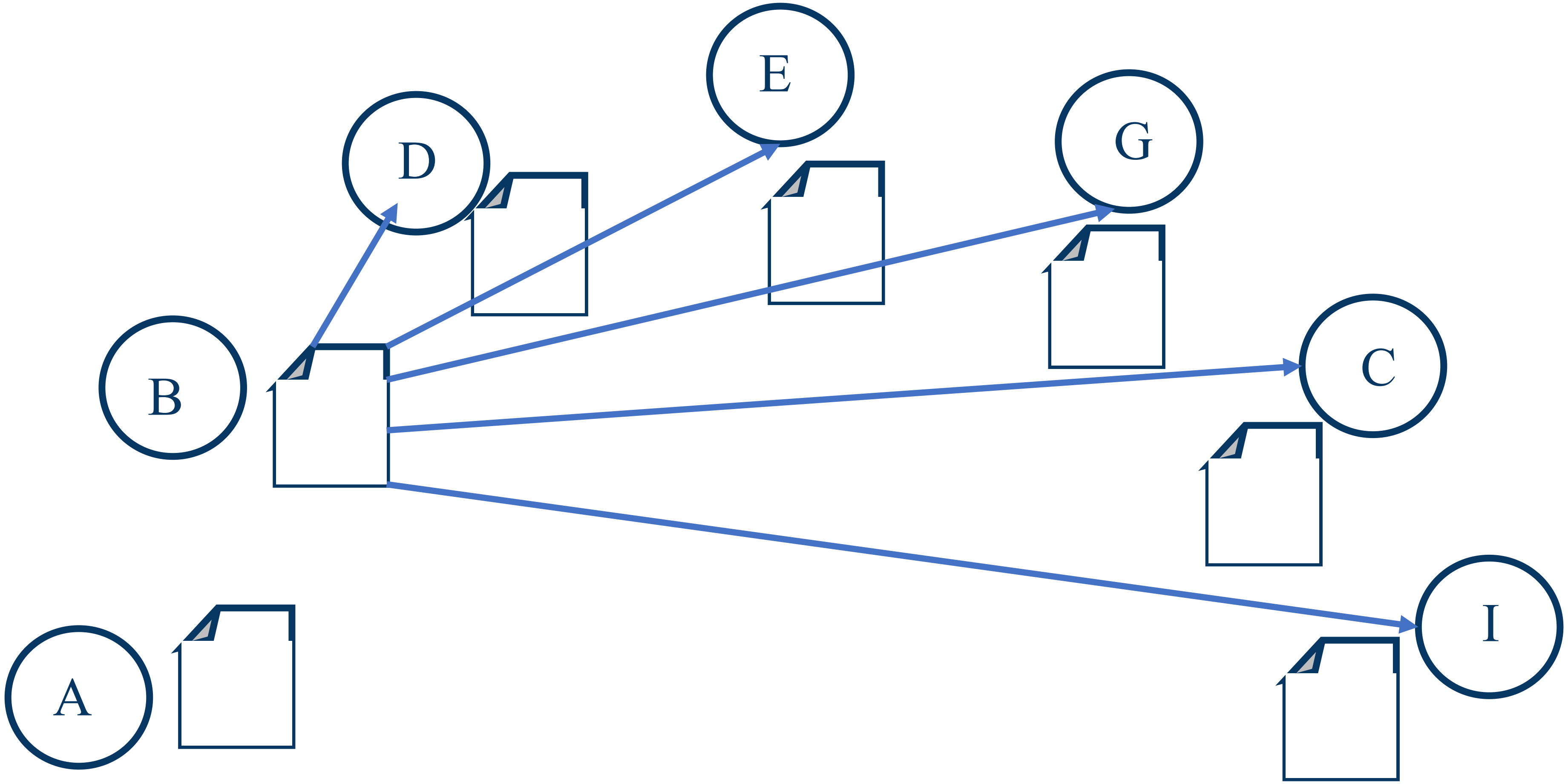
V3

PennState

Alice sends her transaction to Bob

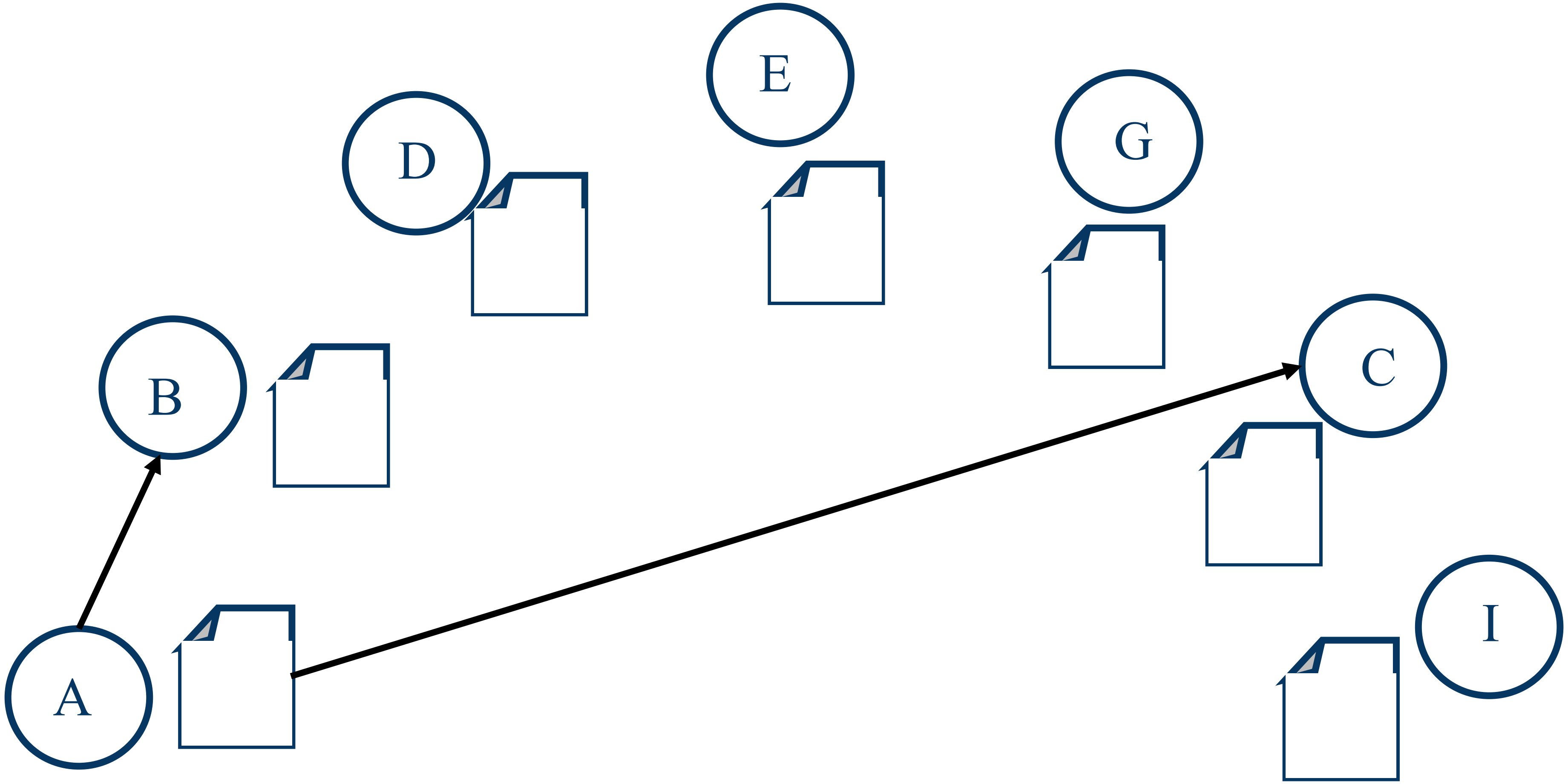


Bob announces the transaction to the world



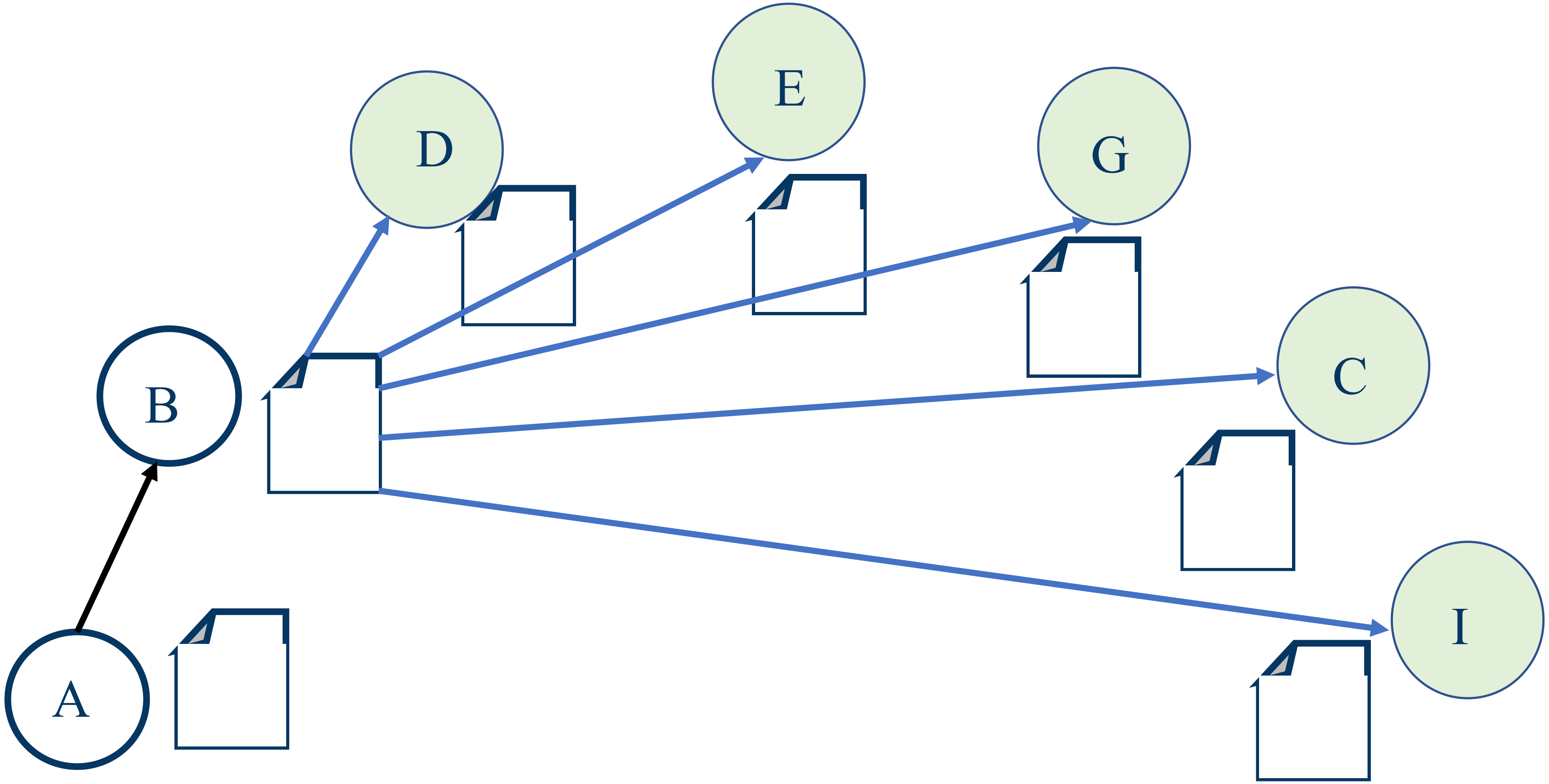
V3

Alice double spends on Bob and Charlie

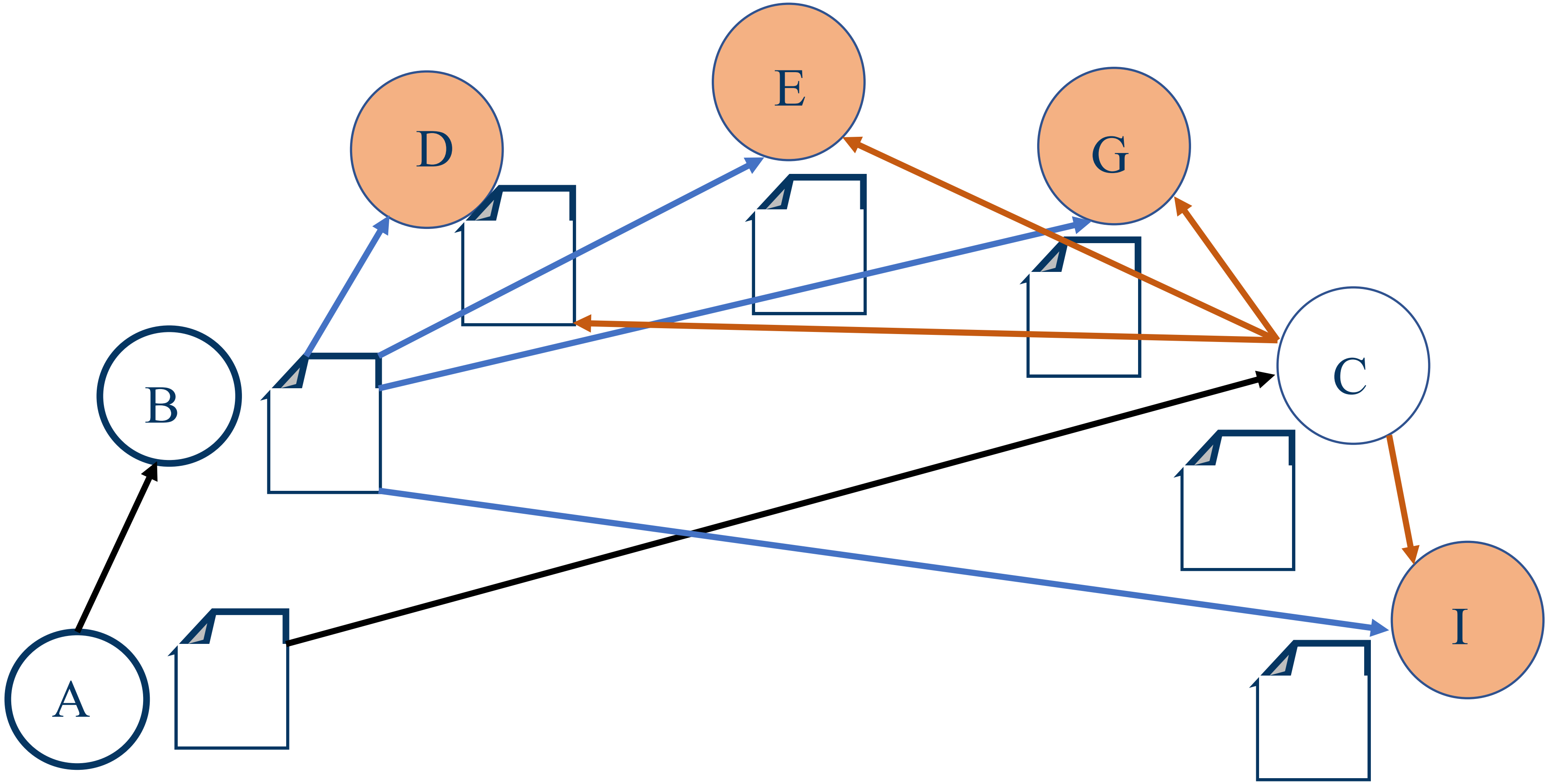


V4

Everyone verifies transactions

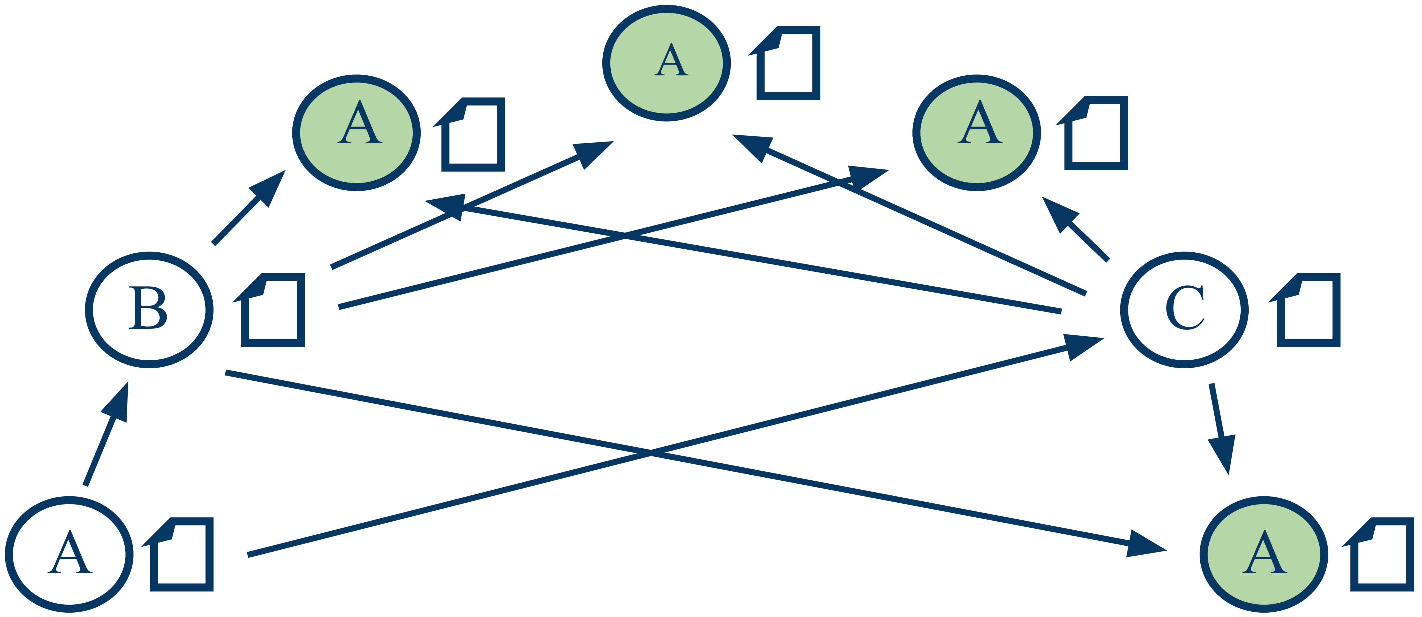


Alice is prevented from double spending



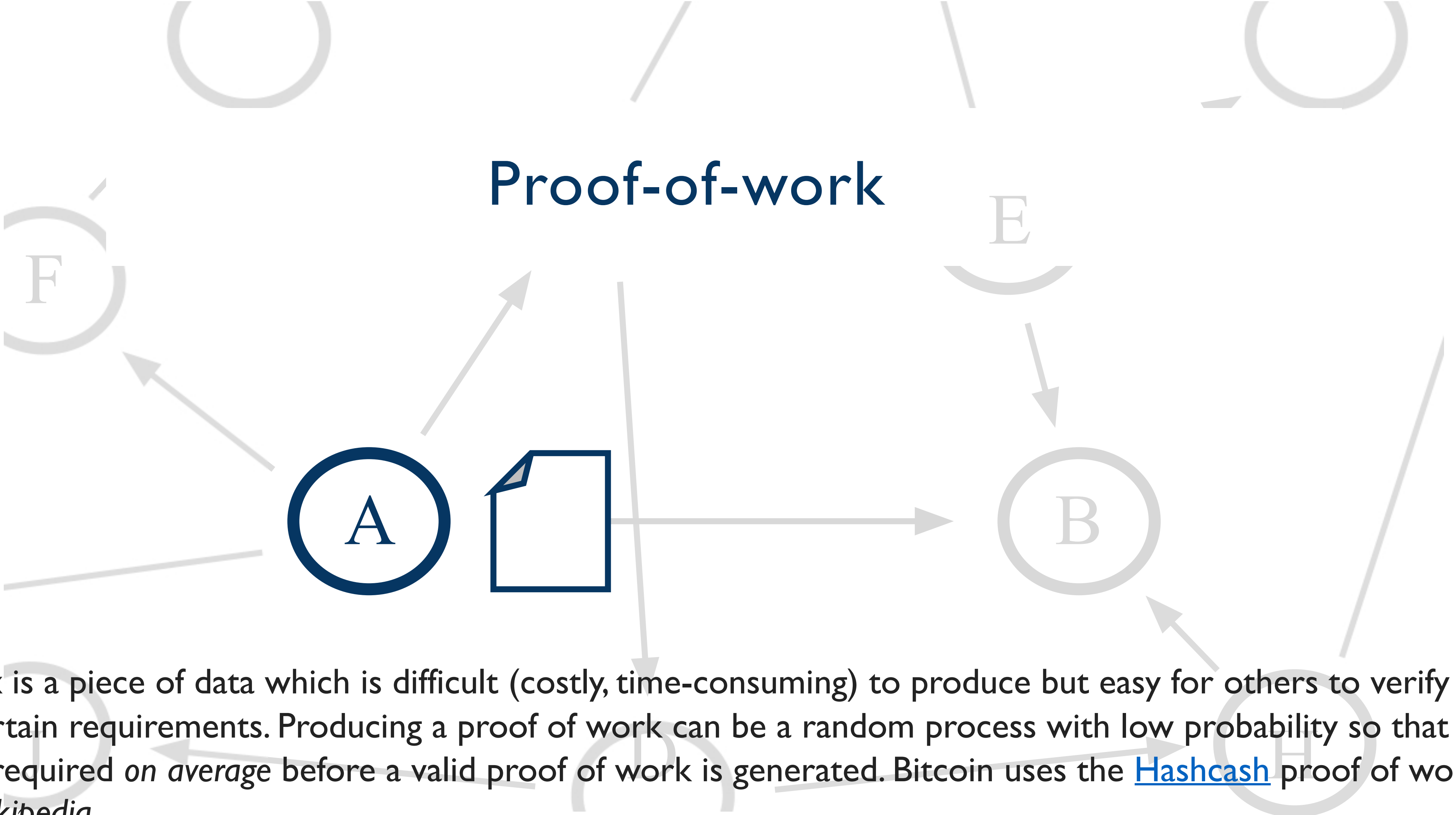
Alice double spends with her multiple identities

Sybil Attack: Creating many fake identities to subvert a system



V5

PennState



A **proof of work** is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements. Producing a proof of work can be a random process with low probability so that a lot of trial and error is required *on average* before a valid proof of work is generated. Bitcoin uses the Hashcash proof of work system. *Source: Wikipedia*

Verifying transactions

1
Check
blockchain

2
Solve
puzzle

3
Announce
block

Proof-of-work as a competition

Summary

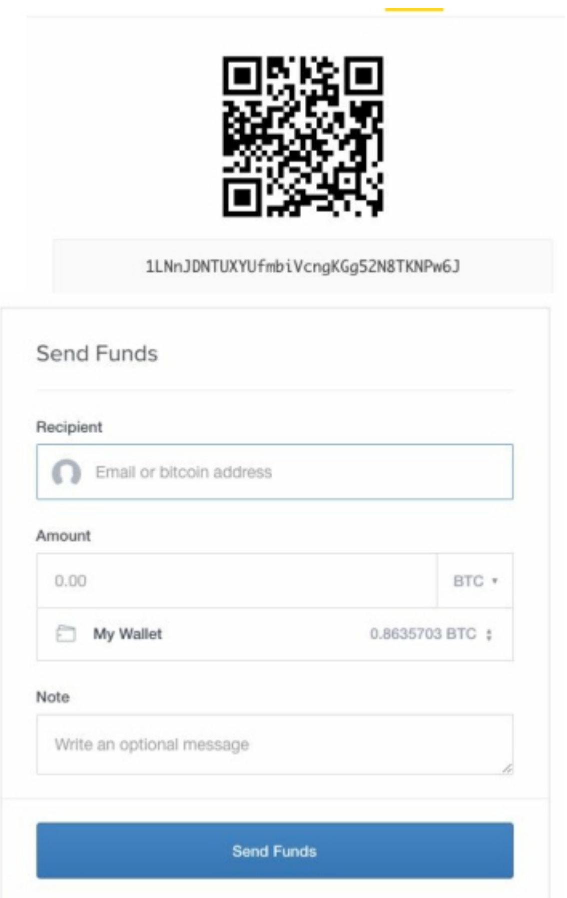
Version	Major feature	Value added
1	Signed messages announced to the network	Basis of entire system
2	Serial numbers	Uniquely identifiable transactions
3	The block chain	Shared record of transactions
4	Everyone verifies transactions	Increased security
5	Proof-of-work	Prevents double spending

Basic Concepts – Identity in Bitcoin

- Send money between pseudonyms
  - pseudonym == address == public key
- Cryptographic primitives
  - digital signature scheme (ECDSA: Elliptic Curve Digital Signature Algorithm)
    - public key/private key pair; like email address + password
  - one-way hash function (SHA-256)
- Bitcoin is hidden in the large amount of public keys
  - Users can generate arbitrarily many key pairs
  - Example Address: 1FtQU9X78hdshngJiCBw9tbE2MYpx87eLT
  - 2^160 possible addresses (1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976 addresses)
  - Grains of sand on earth: 2^63
  - 2^126 is actually only 0.0000000058% of 2^160
  -

Transaction – A Basic Version

- Bitcoin exists as software
  - Transactions are conducted through wallet software
  - Wallet creation generates a Bitcoin address
- To receive money, you share your address
  - Sender specifies address and amount
- The transaction is broadcast to the network, where "miners" verify it and it to the transaction history.



1LNnJDNTUXYUfmbiVcngKGg52N8TKNPw6J

Send Funds

Recipient

Email or bitcoin address

Amount

0.00 BTC

My Wallet 0.8635703 BTC

Note

Write an optional message

Send Funds

Coinbase interface

Mining Sketch

- Notes on Proof-of-Work (PoW)
  - Proof-of-Work is the solution to the mining problem
  - Proof-of-Work is an example of a "Byzantine consensus algorithm"
  - Proof-of-Work is of the a plethora of consensus algorithms
  - Private blockchains tend to use alternative algorithms, but are not completely trustless

- Mining functions as:
  - A minting mechanism that ensure coins are distributed in a fair way
  - An incentive for people to help secure the network
  - Key component that enables you reach consensus in a decentralized currency

Mining Sketch – Finding Blocks

PennState

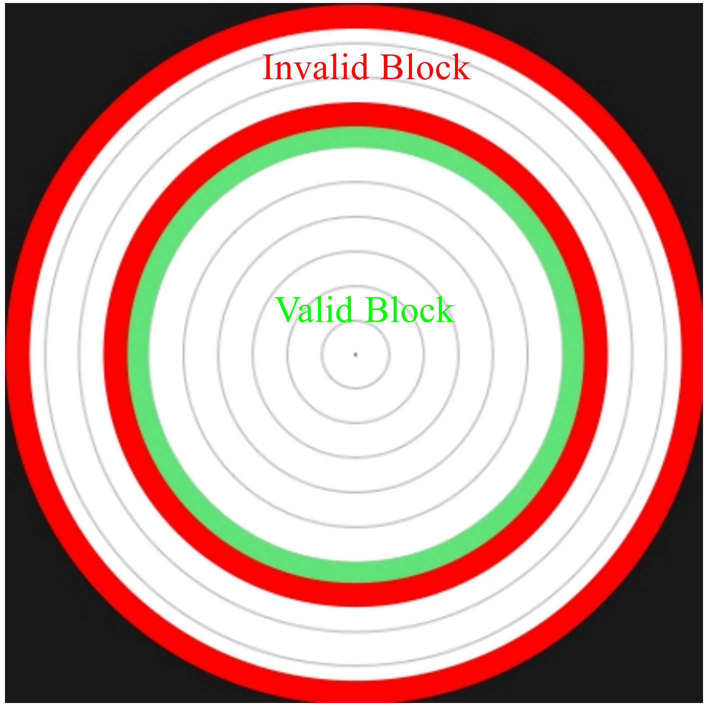
- Finding the PoW => ‘found’ a block; can add block to blockchain
  - Miner who found block adds “**coinbase transaction**”
    - Contains mining reward (currently 12.5 BTC)
  - Miner broadcasts block
  - Other nodes verify, then add to their own copy of the blockchain
- Timeline + stats
  - This happens roughly every 10 minutes
  - Each block can only contain 1MB worth of transactions

What a Miner Does?

- A Bitcoin miner must:
    1. Download the entire Bitcoin blockchain to
    2. Store the entire transaction history
    3. Verify incoming transactions by checking signatures and confirming the existence of valid bitcoins
    4. Create a block using collected valid transactions
    5. Find a valid nonce to create a valid block header (the "mining" part)
    6. Hope that your block is accepted by other nodes and not defeated by a competitor block
    7. Profit!

Block Difficulty: Analogy

- Mining is like throwing darts at a target while blindfolded:
  - Equal likelihood of hitting any ring
  - Faster throwers ⇒ more hits / second
  - Target: within green ring
  - Difficulty inversely proportional to green ring size
    - Green ring adjusts depending on average time to produce valid result
  - If people get better at throwing darts, green circle needs to get smaller



$H(\text{nonce} \,||\, \text{prev\_hash} \,||\, \text{merkle\_root}) < \text{target}$

Block Difficulty: Puzzle Prereqs

- Hash puzzles: the requirement to find a nonce that satisfies the inequality in the lower left region beneath the target: *H(nonce || prev_hash || merkle_root) < target*
- Hash puzzles need to be:
  - 1. Computationally difficult.
    - If finding the proof-of-work requires little work, what's the point?
    - That's why we blindfold the dart-throwers.
  - 2. Parameterizable (variable) cost.
    - Allows for adjustments with global hashrate increases
  - Easily verifiable.
    - Should not be a need for a central authority to verify nonce validity; instead, other miners can rehash the nonce to verify validity.
    - If darts fell out of the dartboard, how can we prove where it hit?

How to Profit From Mining

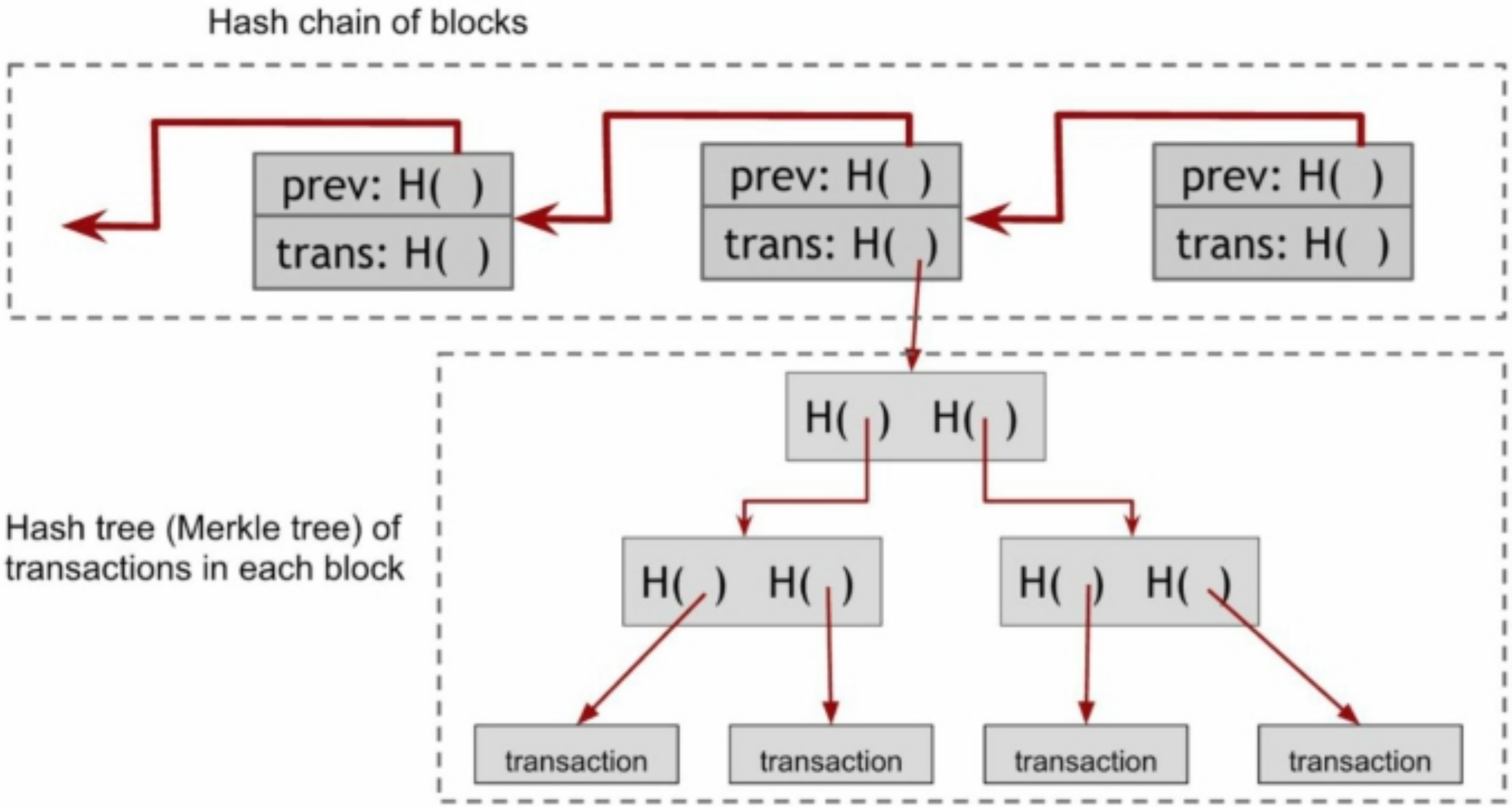
```
MINING_REWARD = BLOCK_REWARD + TX_FEES

MINING_COST = HARDWARE_COST + OPERATING_COSTS

if MINING_REWARD > MINING_COST:
      miner.get_profit()
```

Merkle Tree – Bitcoin Construction

- Transactions are leaves in the Merkle tree, includes a coinbase transaction
- Two hash structures
  - 1. Hash chain of blocks
    - a. These blocks are linked together and based off of each other
      - i. tamper evident
  - 2. A Merkle tree of txs, internal to each block
    - a. Detail: Merkle tree is always full - duplicate the last tx to fill in gaps



Merkle Tree – Mining in More Detail

- Previously, hash of:
  - Merkle Root
  - PrevBlockHash
  - Nonce (varied value) below som value.

- Actually two nonces:
  - In the block header
  - In the coinbase tx

- Hash of
  - PrevBlockHash
  - Coinbase nonce (varied value)
    - Affects the Merkle Root
  - Block header nonce (varied value)

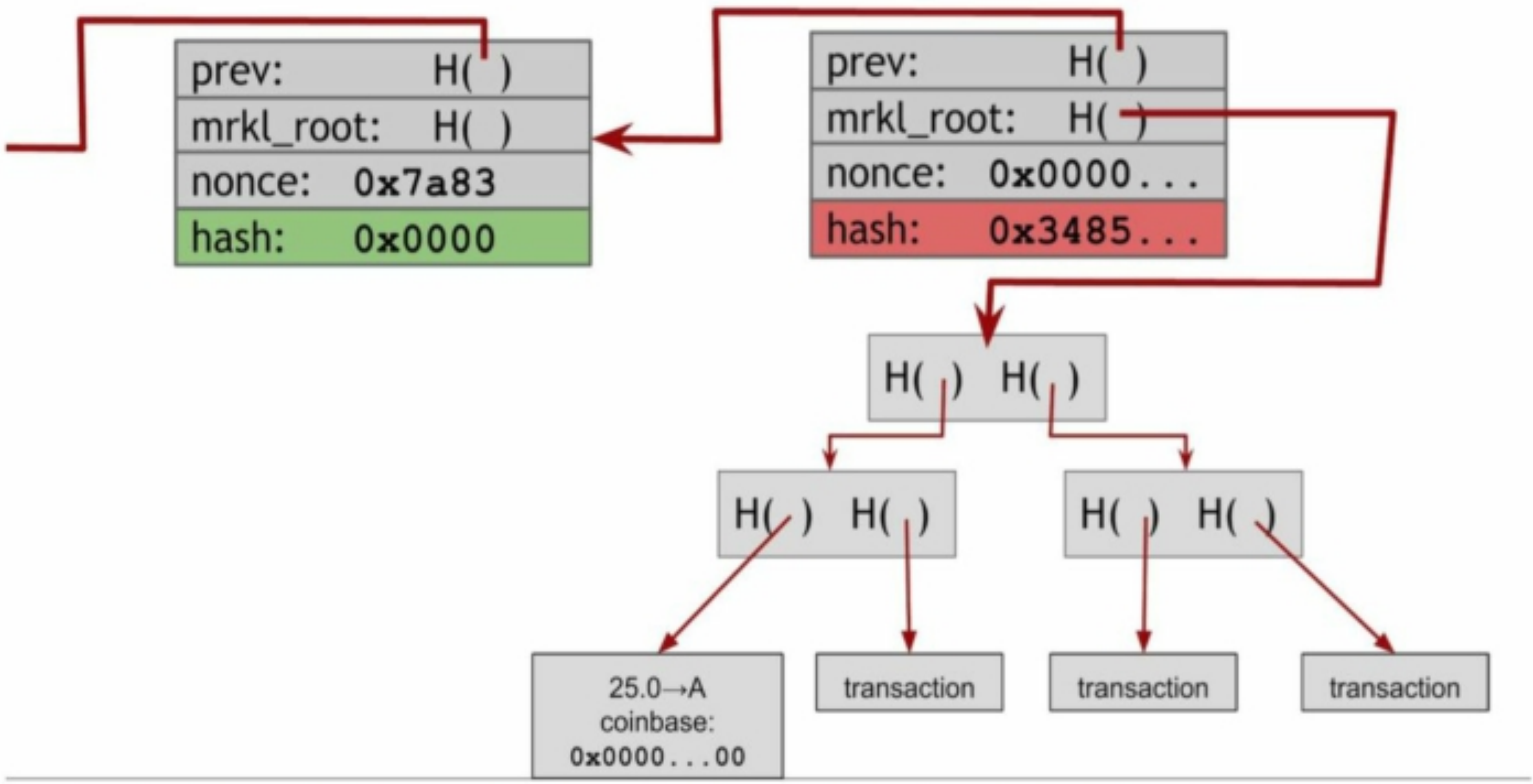


Figure 5.1: Finding a valid block. In this example, the miner tries a nonce of all 0s. It does not produce a valid hash output, so the miner would then proceed to try a different nonce.

51% Attacks

- Major assumptions of Bitcoin
  - No more than 51% pf the network is dishonest
  - An honest majority will always form the longest proof-of-work chain
- 51% Attack attempts to overwhelm the mining power of the network

Thanks

PennState

Thanks to Blockchain At Berkeley for some slides.