SCP: A Computationally Scalable Byzantine Consensus Protocol for Blockchains

Loi Luu, Viswesh Narayanan, Kunal Baweja, Chaodong Zheng, Seth Gilbert, Prateek Saxena

National University of Singapore

Bitcoin doesn’t scale

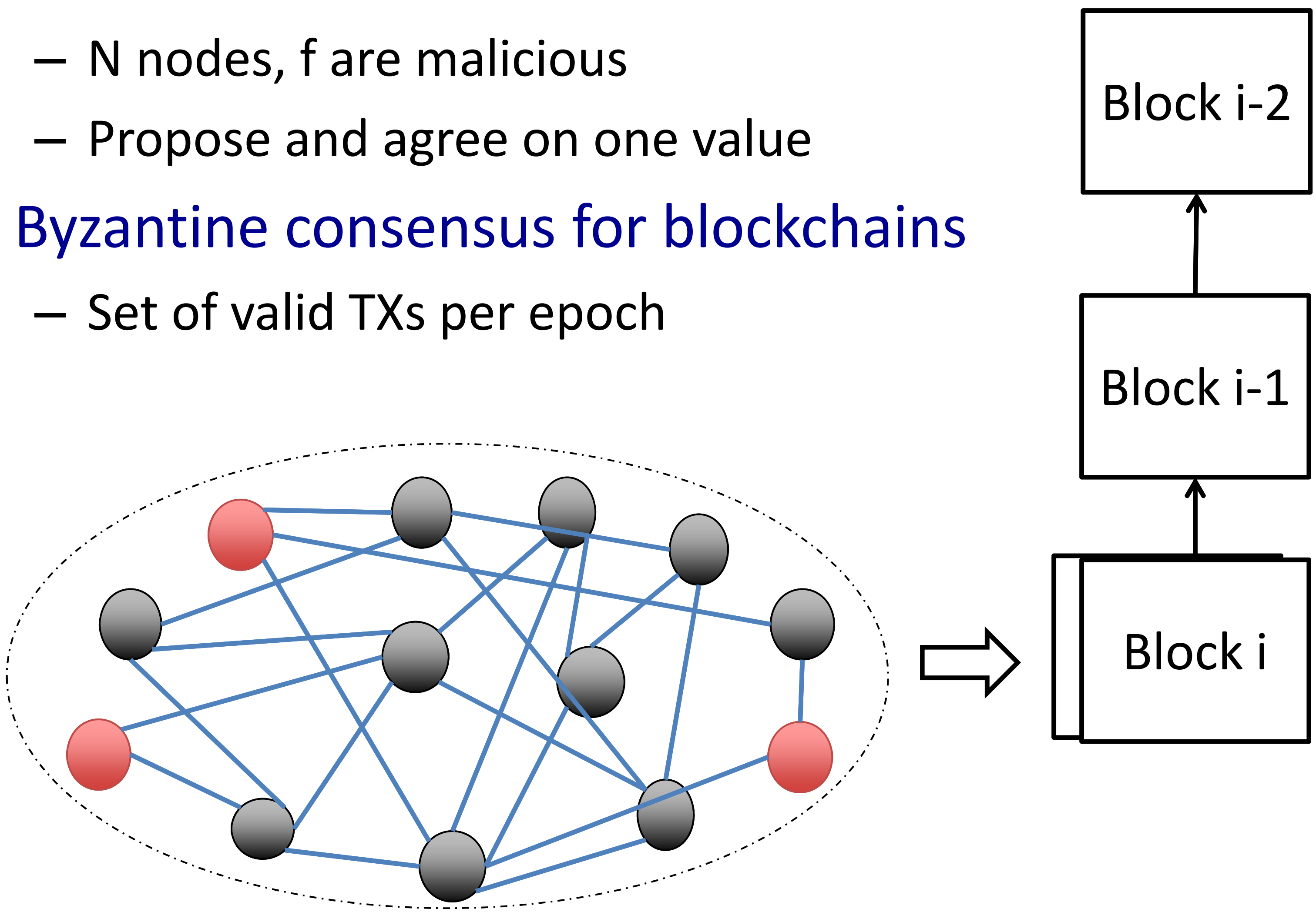
- Hard coded parameters
  - 1 block per 10 minutes
  - 1 MB block size
  - 7 TXs per second
- Today
  - 1-2 TXs per second
  - VISA: 10, 000 TXs per second

Our solution: SCP

- Scale up throughput several orders of magnitude
  - Without degrading any security guarantee
- Several blocks in each epoch
  - No. of blocks ≈ network computation capacity
- Require minimum amount of network bandwidth
  - Broadcast only one block header

Byzantine consensus problem

- Problem
  - N nodes, f are malicious
  - Propose and agree on one value
- Byzantine consensus for blockchains
  - Set of valid TXs per epoch



Classical byzantine consensus protocol

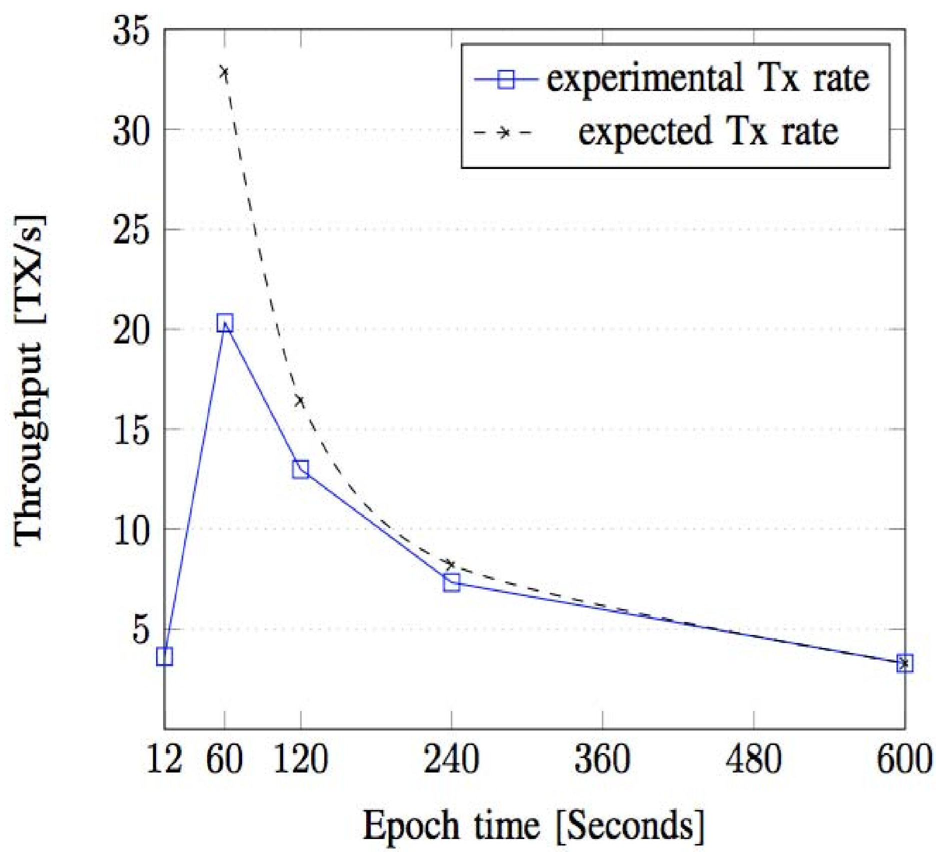
- ✓ Intensive research
  - ✓ Can tolerate $f < n/2$
- x Assumption of known identity set
- x Bandwidth limited
  - – $O(n^2)$ messages (e.g. PBFT)
  - – Work for a small network (e.g $n < 1000$)

Nakamoto consensus protocol

- ✓ Work for network of any size
  - ✓ Select leader by proof of work
- ✓ Linear message complexity
- x Does not scale well in practice
  - x One block per epoch
  - x Bandwidth = O(block size)
  - x Reparameterization is not a long term solution

Reparameterization: reducing epoch time

- Setup
  - Using Amazon EC2
  - Run over 5 regions
- Results
  - TX rate increases until some threshold
  - <u>Drops at 12 second epoch time</u>



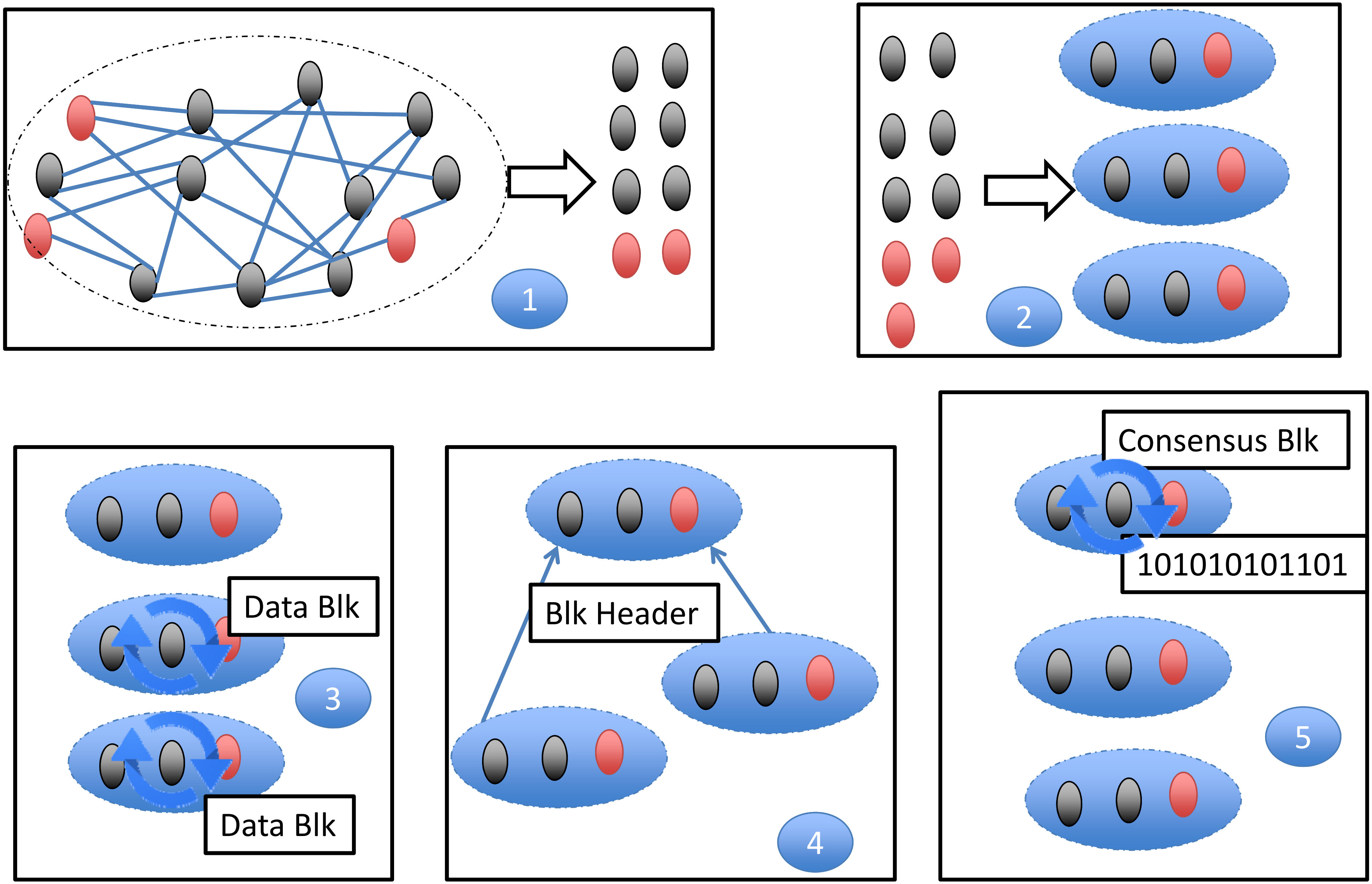
Problem

- Secure & scalable consensus protocol
  - Compete with VISA?

SCP overview

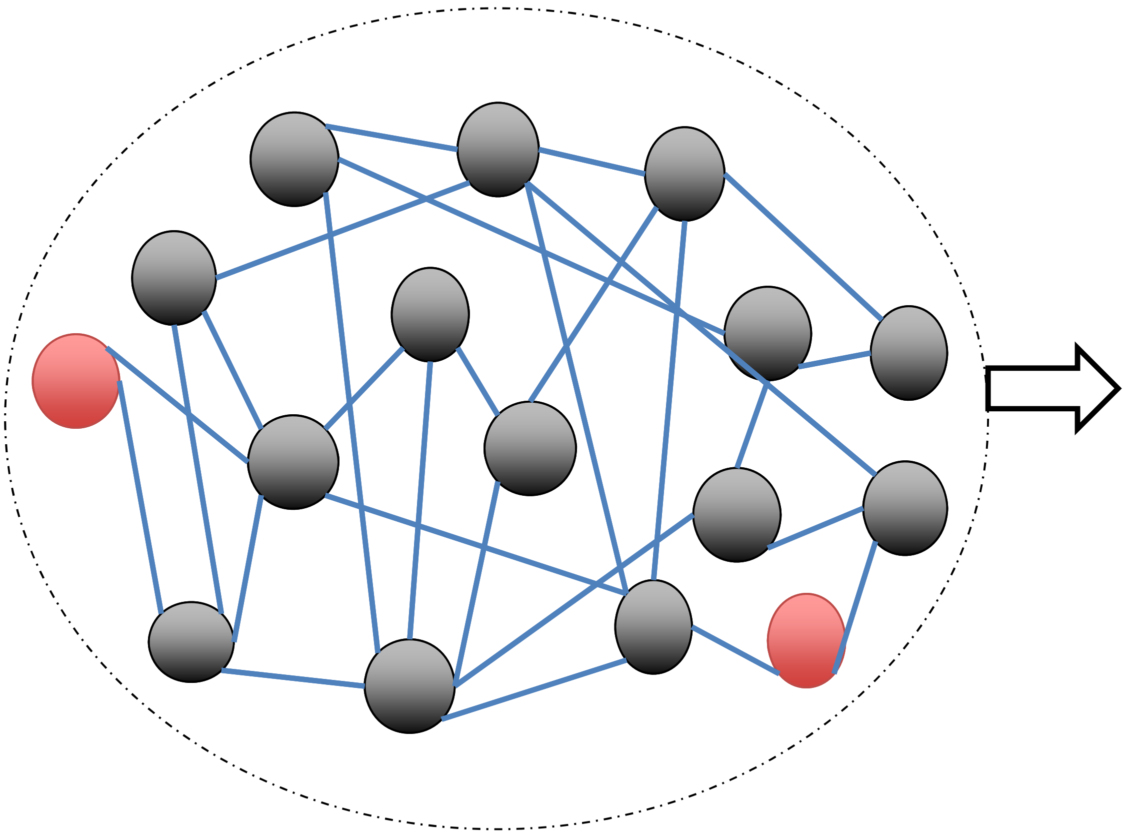
- Adjust throughput based on network mining power
  - Split the network into several committees
  - Committees propose blocks in *parallel*
  - No. of committees ≈ F(network mining capacity)
- Data needed for reaching consensus is minimal
  - Consensus data != transactional data
  - Verify block without block data
  - Selectively download block data

SCP protocol



Step 1: Identity establishment

- Solve PoW
  - SHA2(EpochRandomness || IP || pubkey || nonce) < D

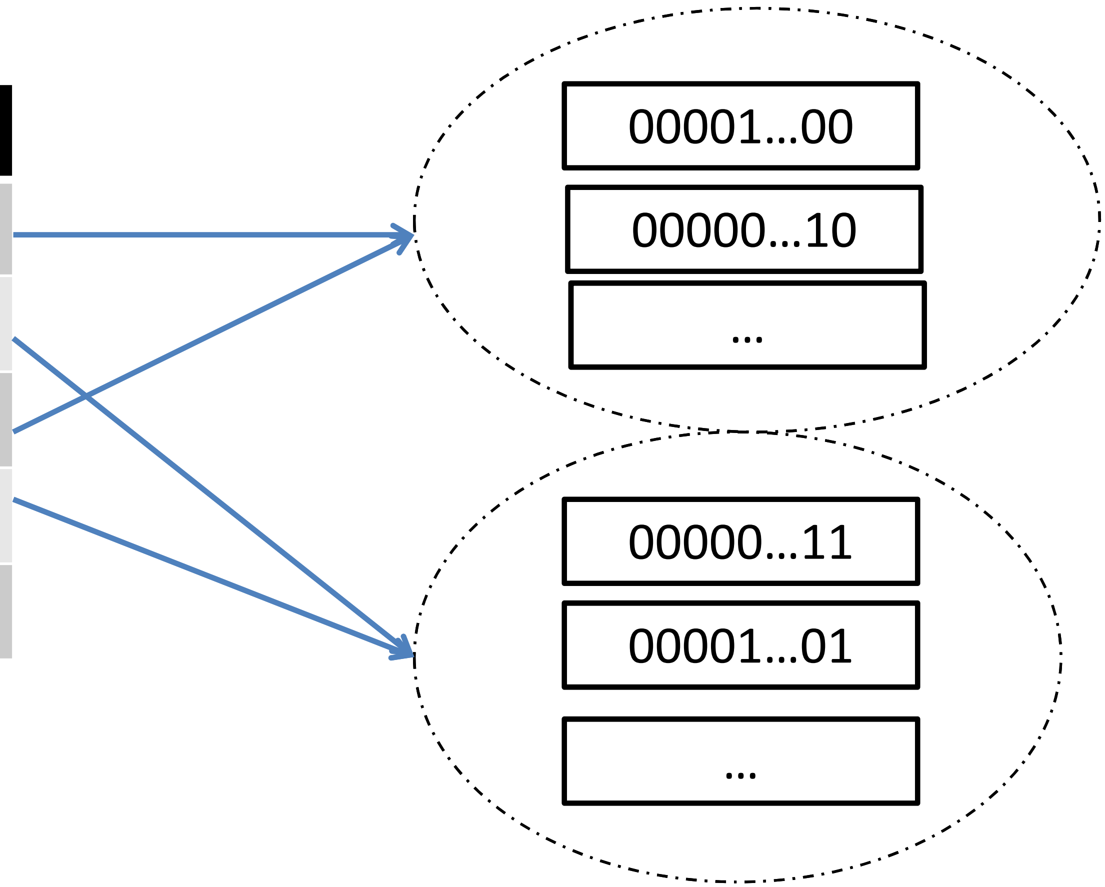


ID	PoW	IP	Pubkey
1	00001…	a.b.c.d	ABC…
2	00001…	a.b.c.e	DEF…
..	…	…	….

Step 2: Assigning committees

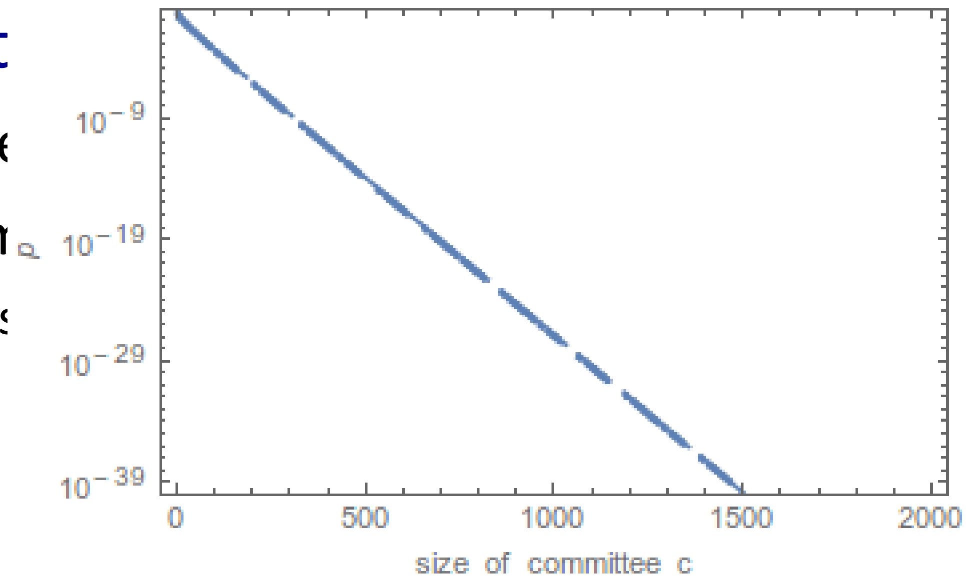
- Randomly & uniformly distribute identities to committees
  – Based on the last k bits of PoW

ID	PoW
1	00001…00
2	00000…01
3	00000…10
4	00001…11
..	…



Size of a committee C

- Decide the probability of majority honest
  - P(error) reduces exponentially with C
    - f = N/3, C = 400, p(error) ≈ 10^{-12}
    - f = N/3, C = 100, p(error) ≈ 0.0004
- Why majority honest wit
  - Run practical authenticate
  - Allow others to verify con
    - At least 1 member is hones

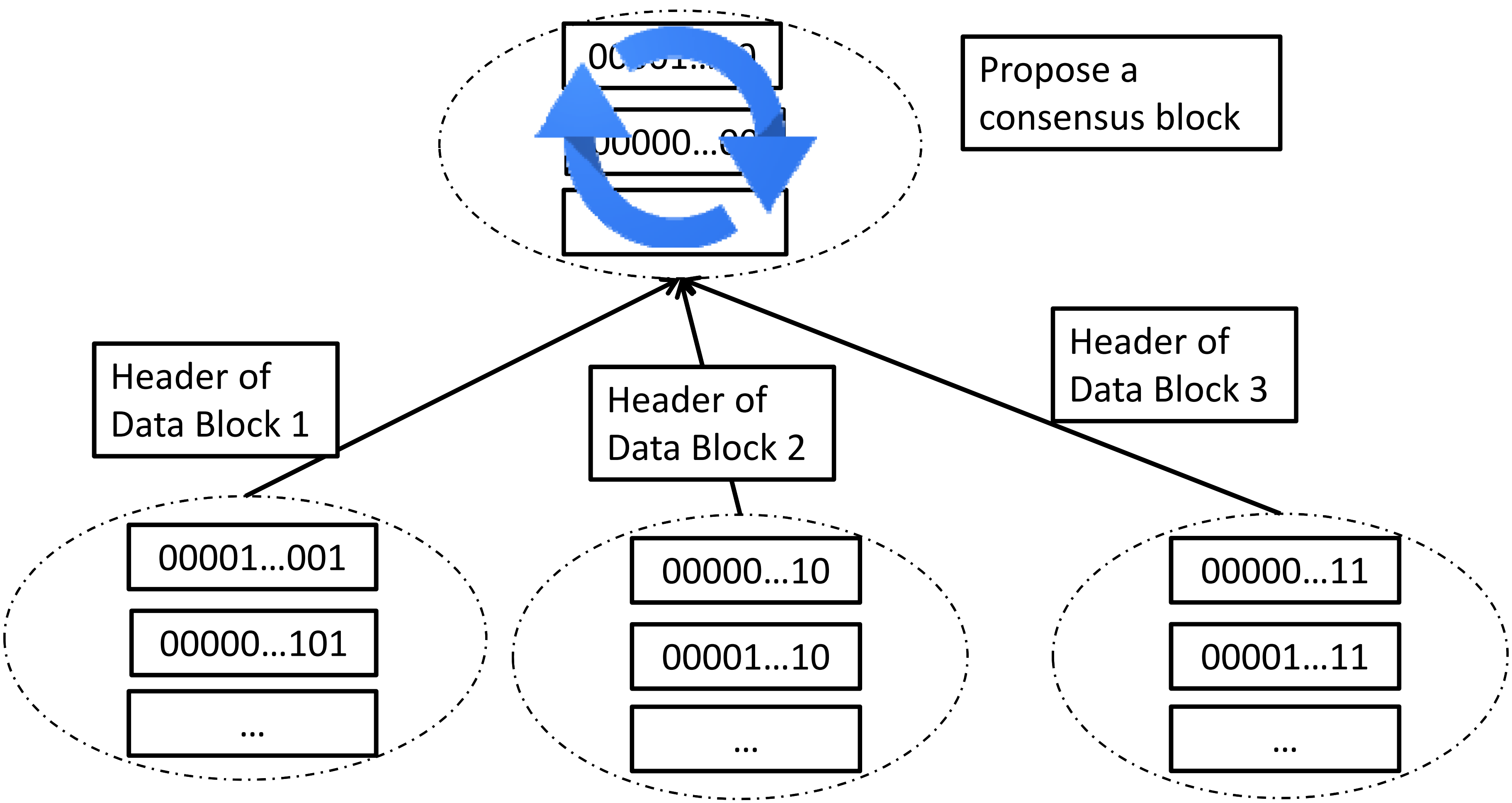


Step 3: Propose a block within a committee

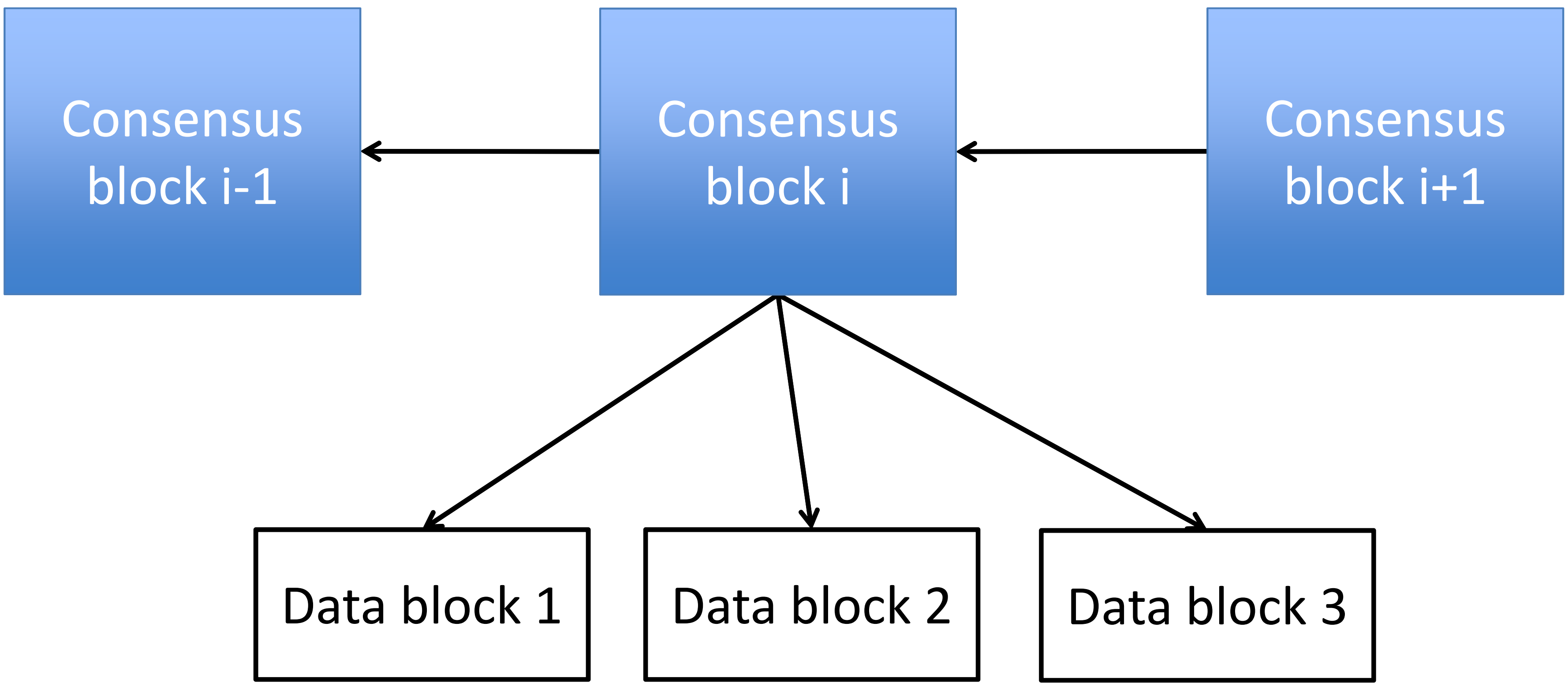
- Run a classical Byzantine consensus protocol
  - Members agree & sign on one valid data block
  - No. of messages ≈ $O(C^2)$
- TX sets included in data blocks are disjoint
  - Include TXs with a specific prefix

Block	TX's IDS
Data Block 1	00…
Data Block 2	01…
Data Block 3	10…
Data Block 4	11…

Step 4: Final committee unions all results

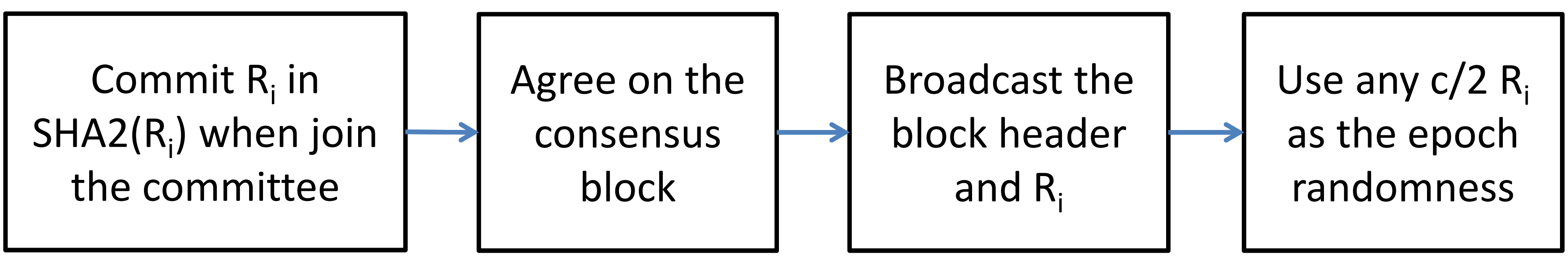


SCP blockchain



Step 5: Generate an epoch randomness

- Goal
  - Generate a fresh randomness
  - Adversary cannot control or predict
- Common approach: Use consensus block hash
  - Problem: adversary can predict the consensus block early
- Our approach: Users can have different randomness

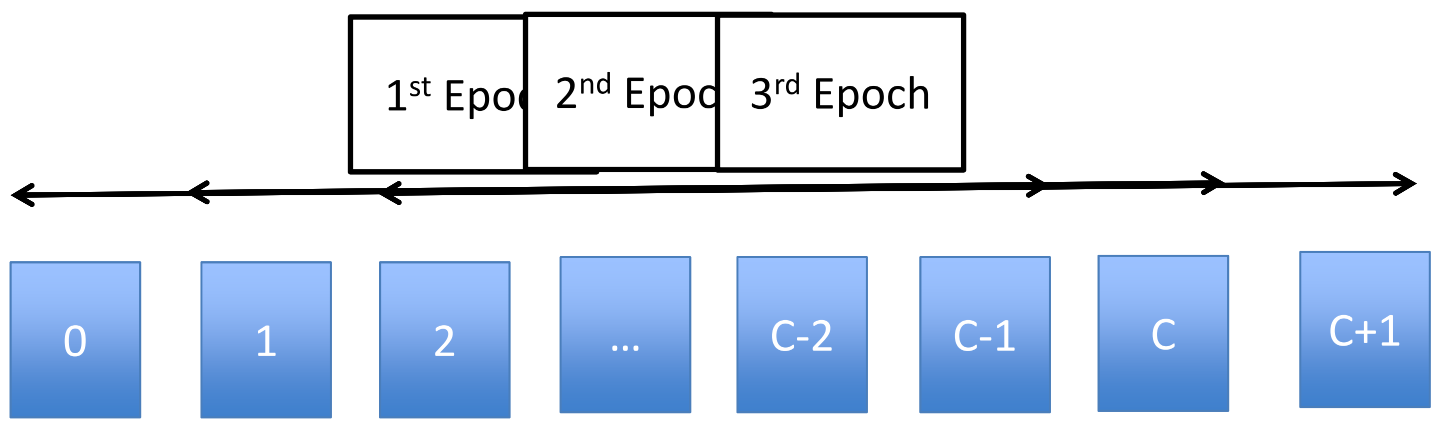


Implement a SCP-based cryptocurrency

- Challenges
  - How to form committees efficiently
    - Too many new identities in each epoch
    - Epoch time may be long to prevent conflict
  - Double spending transactions
    - Without previous block data?

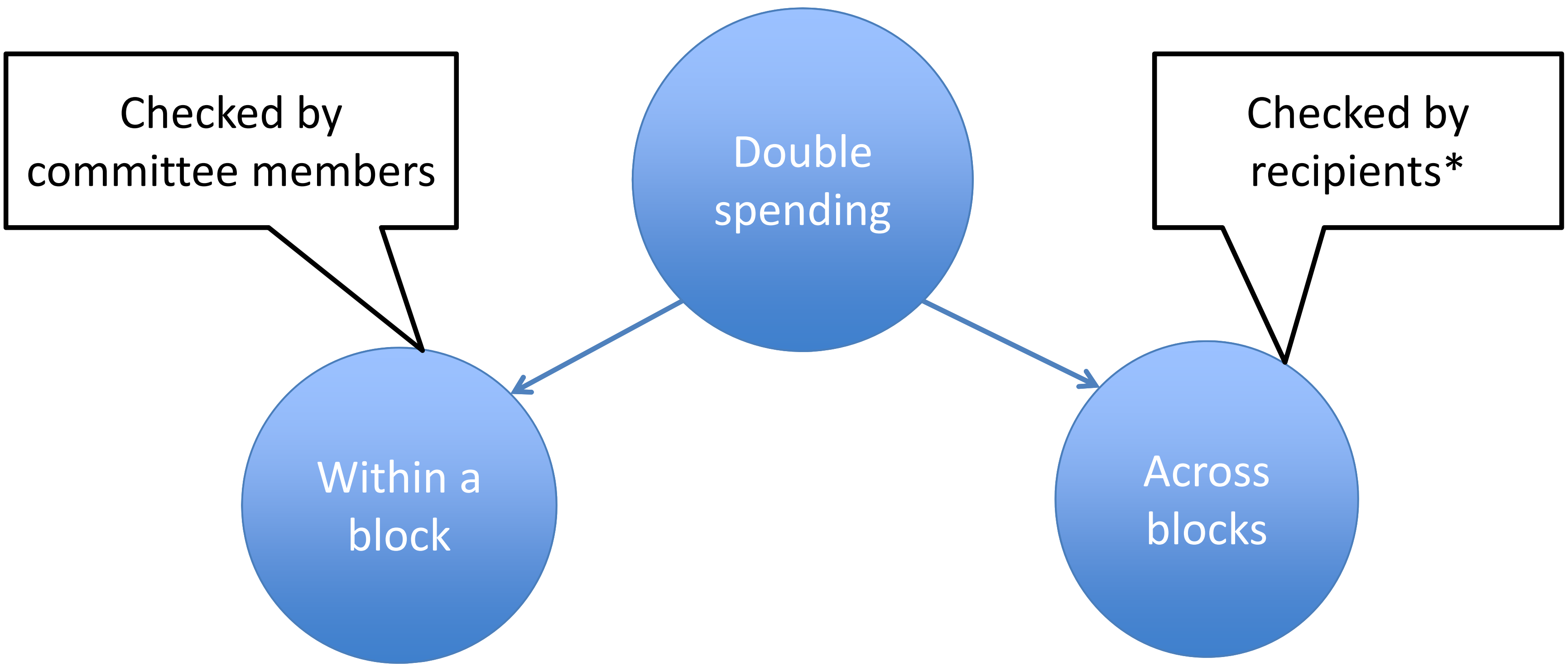
Forming committees efficiently

- Approach: Reuse identities from previous epoch
  - Elect one new member and remove the oldest one
  - Number of new identities ≈ number of committees



Avoid double spending

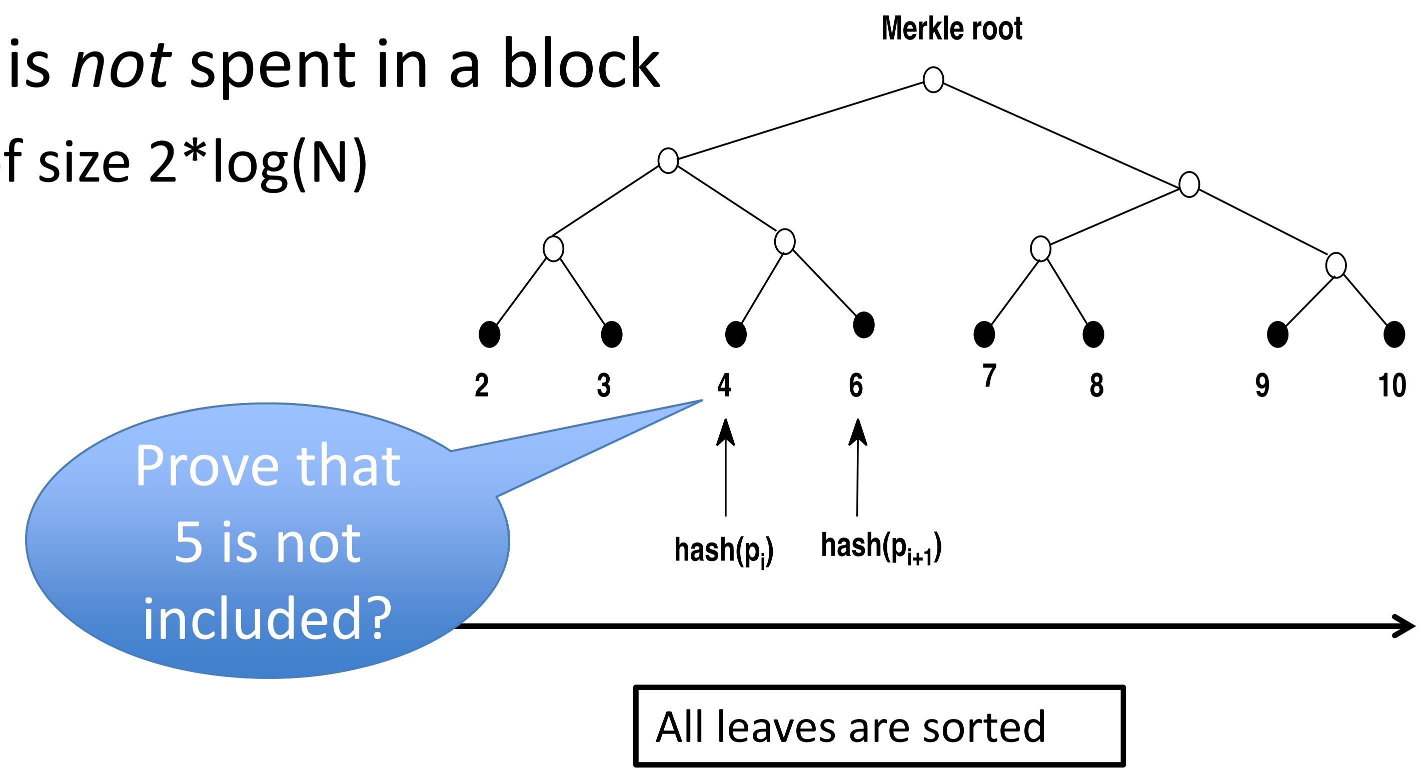
- Approach:
  - Split double spending check into both miners and users (recipients)



*: [Proof-of-publication]

Checking double spending across blocks

- Merkle tree of TX inputs
  - An input is spent in a block
    - Proof of size log(N)
  - An input is *not* spent in a block
    - Proof of size 2*log(N)



Checking double spending across blocks (2)

- Sender proves that the TX's input is not spent elsewhere
  - The proof of size L*log(N)
  - Can be optimized
- Recipient checks by using only consensus block headers
  - Actively support SPV clients without a trusted third party
  - Support 1-confirmation TXs

Conclusion

- SCP scales almost linearly with network mining capacity
  - More mining power, higher transaction rate
  - Reduced network bandwidth
  - Secure
- Applicable to several applications
  - Cryptocurrency, decentralized database, etc

Q&A

Loi Luu
loiluu@comp.nus.edu.sg
www.comp.nus.edu.sg/~loiluu

Future work

- Incentive structure
  – Incentivize committee members and other parties
- Prevent DoS attack by sending invalid TXs
  – Users can send arbitrary TXs to the blockchain now
- Rollback solution
  – P(error) != 0

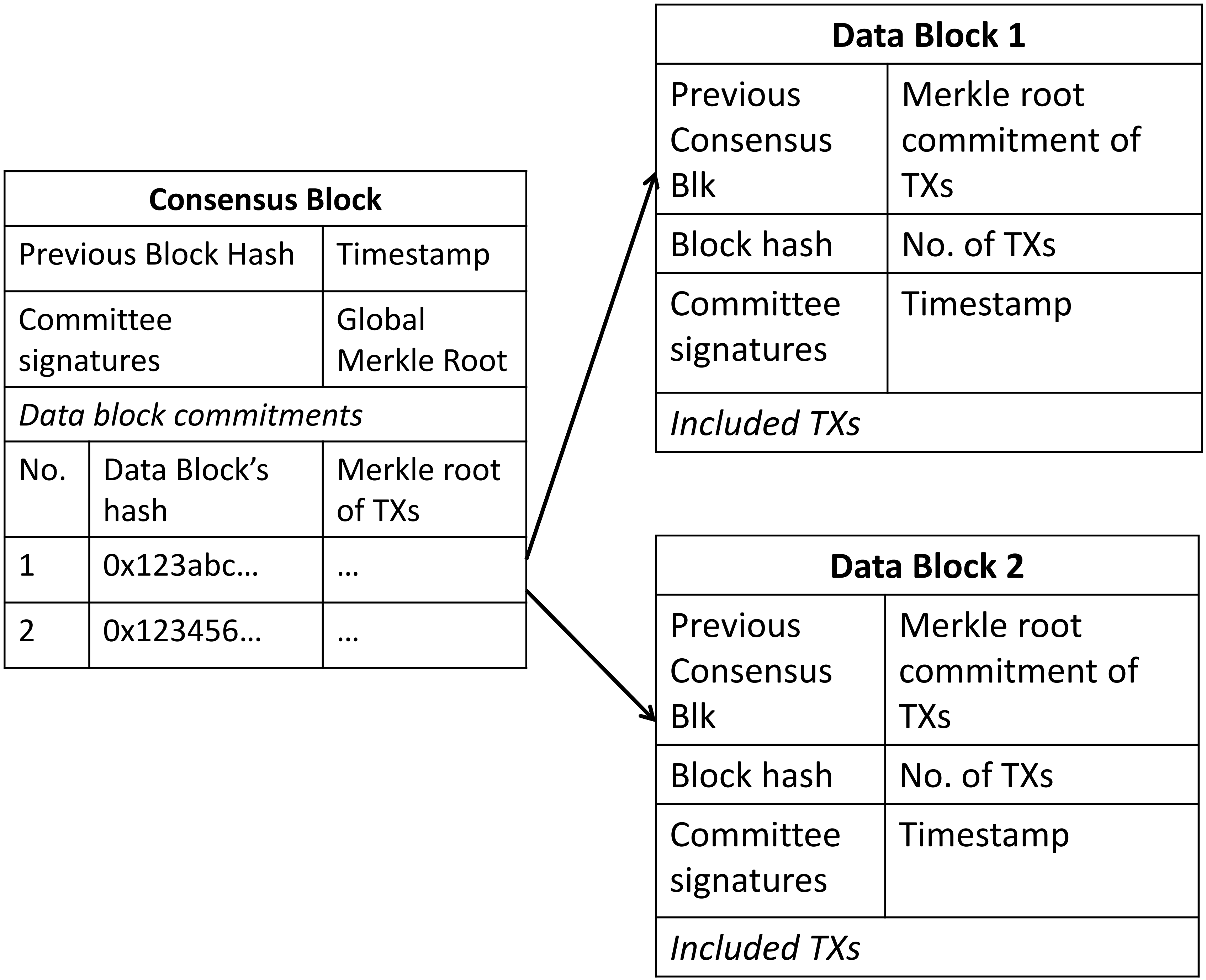
Related work

- Bitcoin-NG & Ghost
  - ✓ Allow more blocks
  - x Does not separate consensus plane and data plane
- Lighting network
  - ✓ Allows more micro transactions
  - x Does not solve scalability problem
- Sidechains
  - ✓ Good for experimenting new blockchains
  - x Does not make Bitcoin scalable

Adjusts number of committees frequently

- Similar to how Bitcoin adjusts the block difficulty
  - T: the expected epoch time
  - T’: the averaged epoch time of the most 1000 recent blocks
  - S: Current number of committees
  - S’: adjusted number of committees

$$S' \log(S') = S \log(S) \frac{T}{T'}$$


Consensus Block
Previous Block Hash
Timestamp
Committee signatures
Global Merkle Root
Data block commitments
No.
Data Block’s hash
Merkle root of TXs
1
0x123abc…
…
2
0x123456…
…
Data Block 1
Previous Consensus Blk
Merkle root commitment of TXs
Block hash
No. of TXs
Committee signatures
Timestamp
Included TXs
Data Block 2
Previous Consensus Blk
Merkle root commitment of TXs
Block hash
No. of TXs
Committee signatures
Timestamp
Included TXs

SCP properties

- Number of data blocks ≈ network mining power
  - Frequent adjustment of no. of blocks
- Data broadcast to the network is minimal
  - Broadcast data is independent of block size
- Secure against adaptive adversary w.h.p.
  - Can reparameterize c to secure against stronger adversary