

Algorand: Scaling Byzantine Agreements for Cryptocurrencies

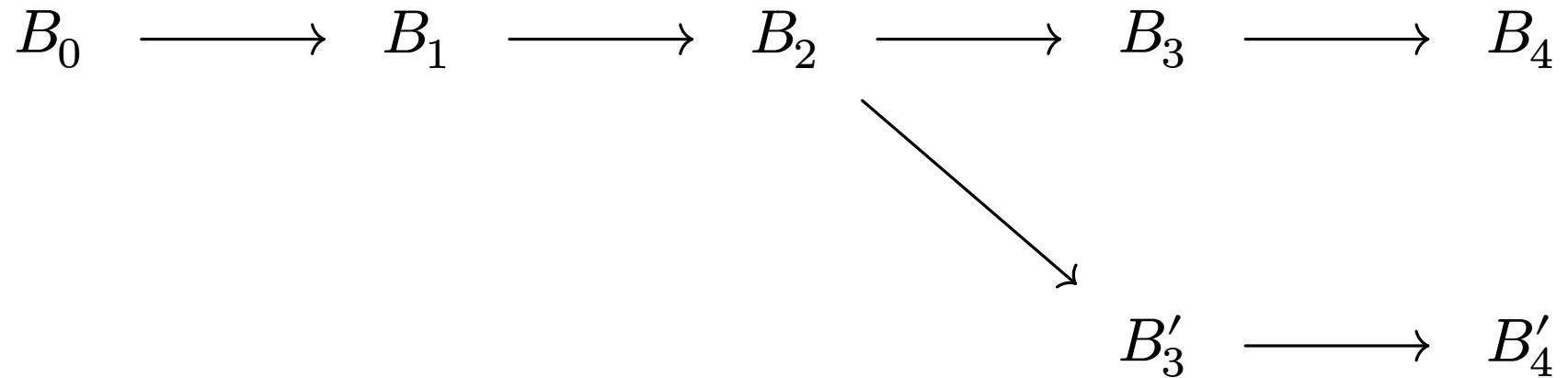
Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, Nickolai Zeldovich
MIT CSAIL

SOSP'17

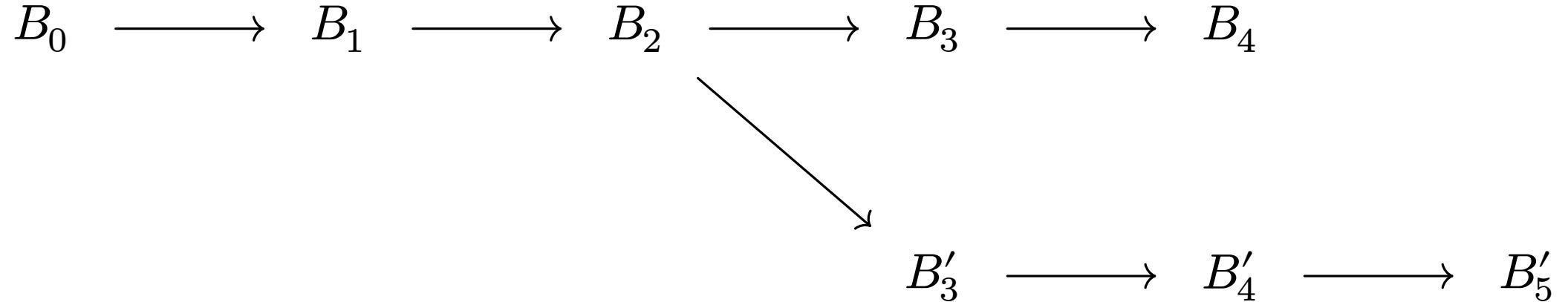
Context Introduction

- Cryptographic currencies
- Avoiding centralized authorities
- Trade-off between latency and confidence
- Double spending problem

Nakamoto consensus & Proof of Work



Nakamoto consensus & Proof of Work



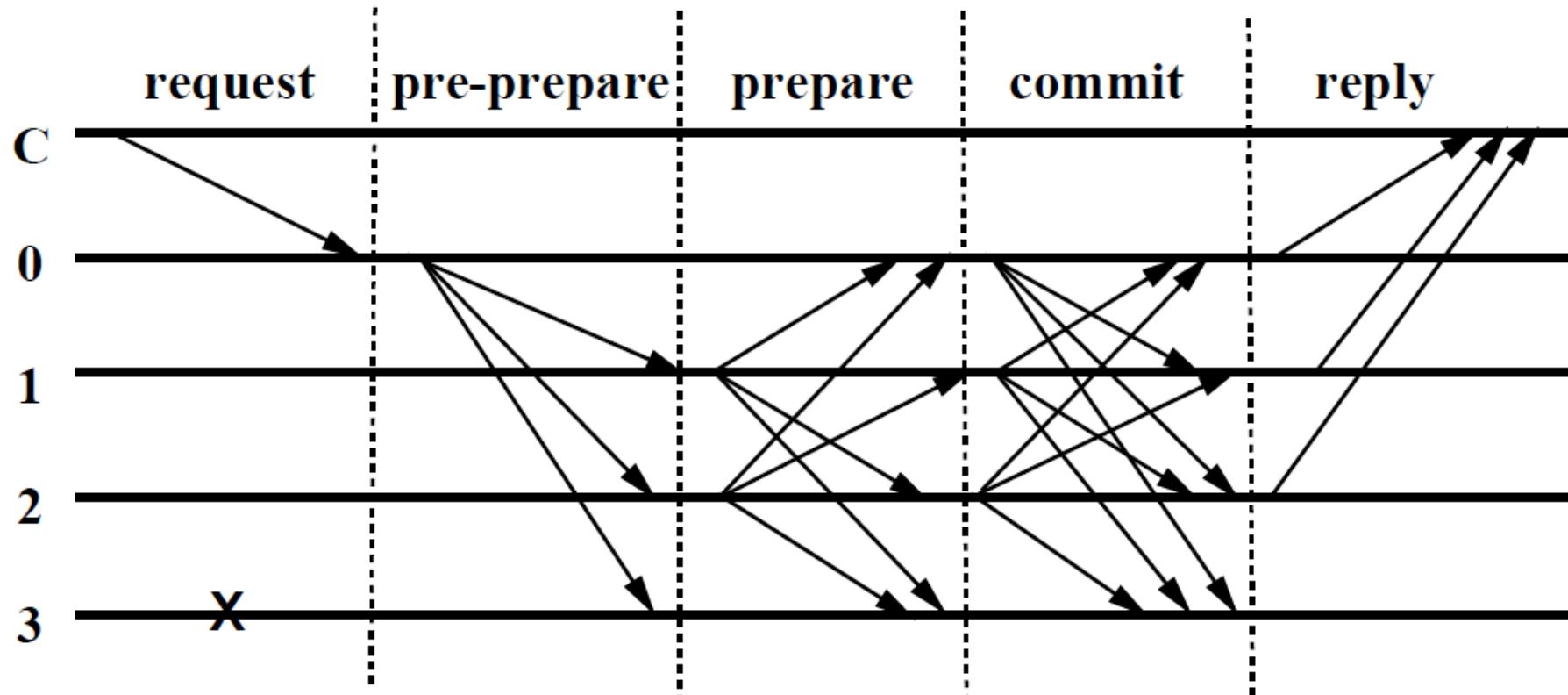
Nakamoto consensus & Proof of Work

- No confident commit
- Possible forks
- Latency problem
- Scalability

Byzantine Consensus

- Predefined set of servers
- Denial of service attack
- All to All communication
 - ▶ Bad Scalability

Example: PBFT



Algorand

- New cryptocurrency
- Confirmation in order of minute
- Scalable (No all to all communication)

Algorand: Network structure

- Dynamic size network
- Scalable (No all to all communication)
- No predefined set of committee

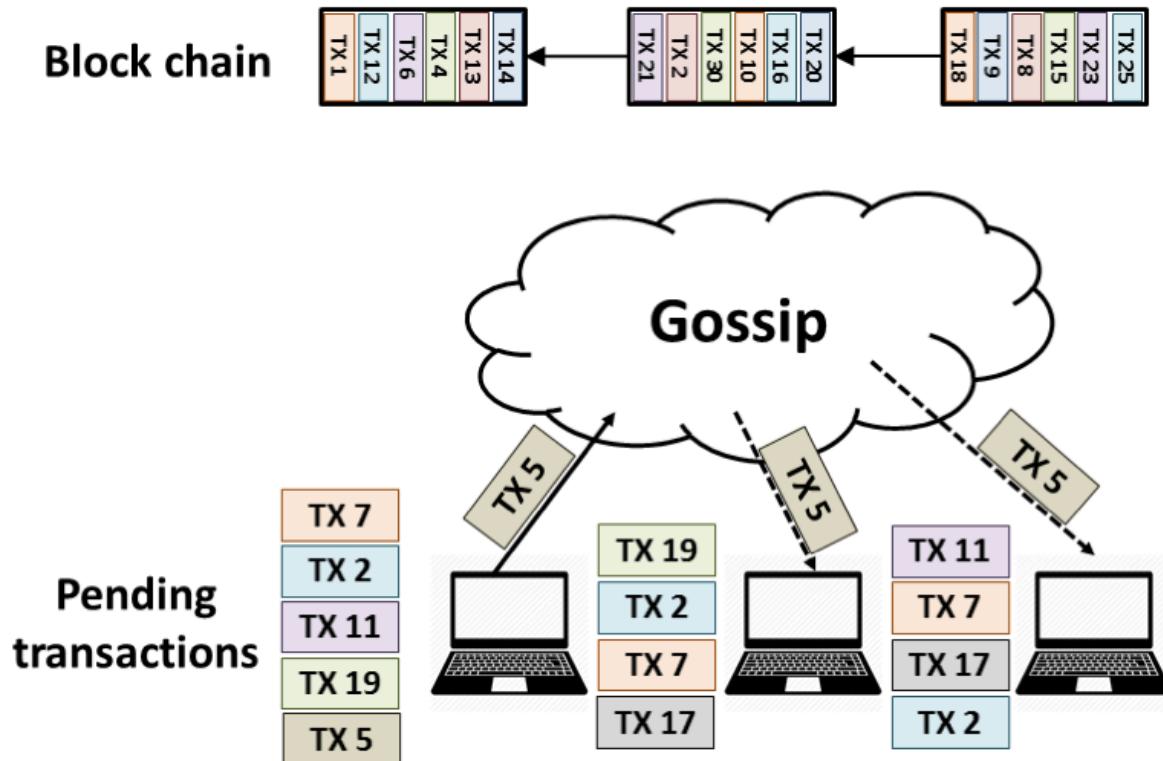
Algorand: BA*

- Proof of Stake
 - Fraction of the money held by honest users is at least a constant greater than $2/3$.
- Confirmation in order of minute
- No predefined set of committee

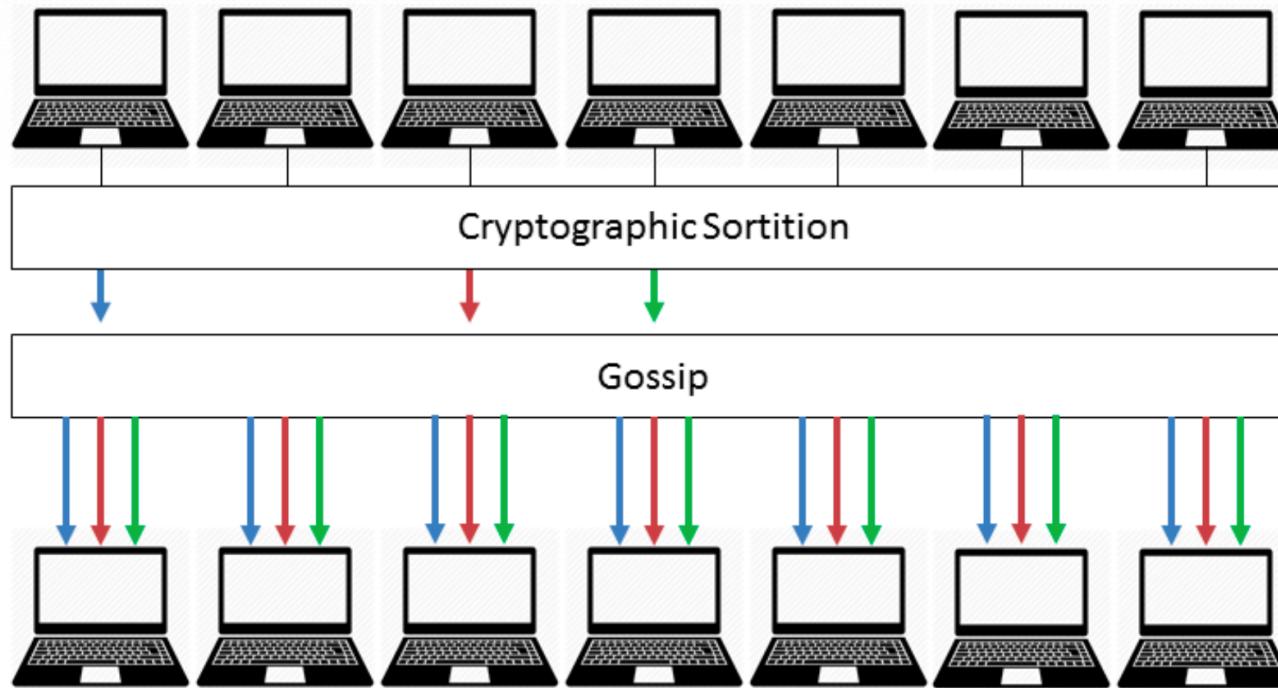
Algorand: Key components

- Gossip Network
- Cryptographic sortition (for choosing small committee)
- BA*

Algorand: Gossip



Algorand: Block Proposal



Algorand: Sortition

procedure Sortition($sk, seed, \tau, role, w, W$):

$\langle hash, \pi \rangle \leftarrow \text{VRF}_{sk}(seed || role)$

$p \leftarrow \frac{\tau}{W}$

$j \leftarrow 0$

while $\frac{hash}{2^{hashlen}} \notin \left[\sum_{k=0}^j B(k; w, p), \sum_{k=0}^{j+1} B(k; w, p) \right]$ **do**
 └ $j++$

return $\langle hash, \pi, j \rangle$

Algorand: Sortition Verification

```
procedure VerifySort( $pk, hash, \pi, seed, \tau, role, w, W$ ):  
    if  $\neg VerifyVRF_{pk}(hash, \pi, seed || role)$  then return 0;  
     $p \leftarrow \frac{\tau}{W}$   
     $j \leftarrow 0$   
    while  $\frac{hash}{2^{hashlen}} \notin \left[ \sum_{k=0}^j B(k; w, p), \sum_{k=0}^{j+1} B(k; w, p) \right]$  do  
         $\quad j++$   
    return  $j$ 
```

Algorand: Evaluation