Heterogeneous Trust with Probabilistic Witnesses

Liron Schiff[1]

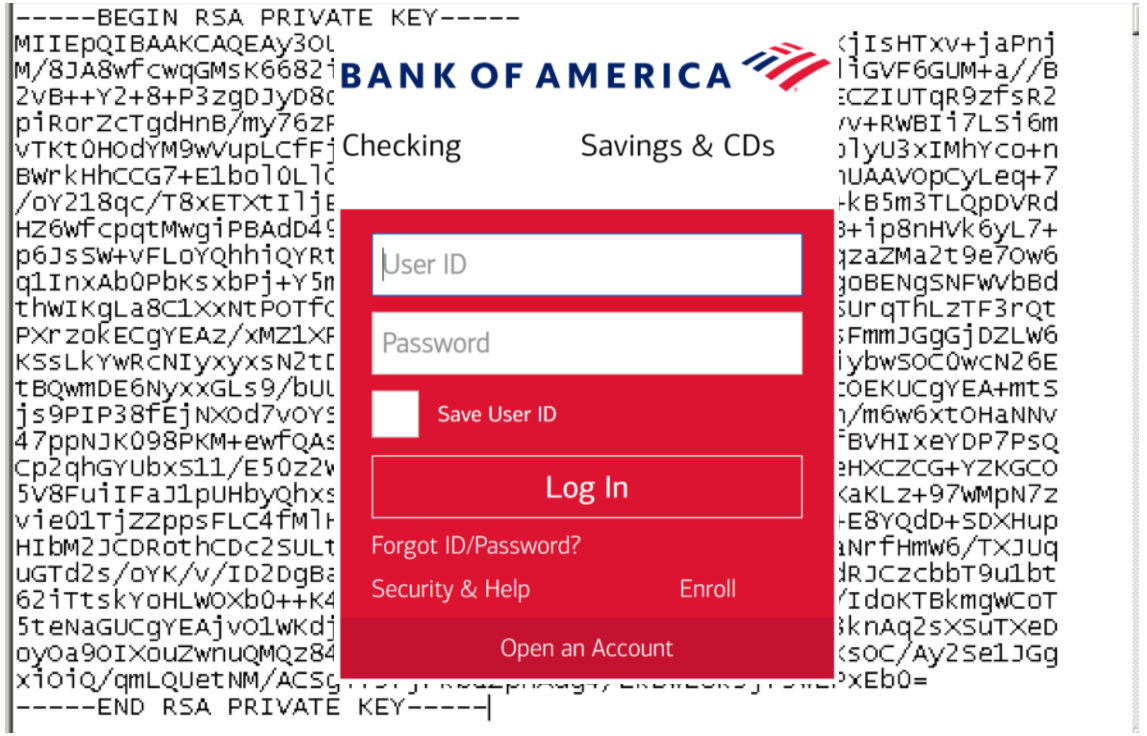
Joint work with:

Veronika Anikina[2], João Paulo Bezerra[3,4], Petr Kuznetsov[3,4] and Stefan Schmid[5]

(1) Akamai (2) ITMO UNIVERSITY (3) TELECOM Paris (4) INSTITUT POLYTECHNIQUE DE PARIS (5) TECHNISCHE UNIVERSITÄT BERLIN

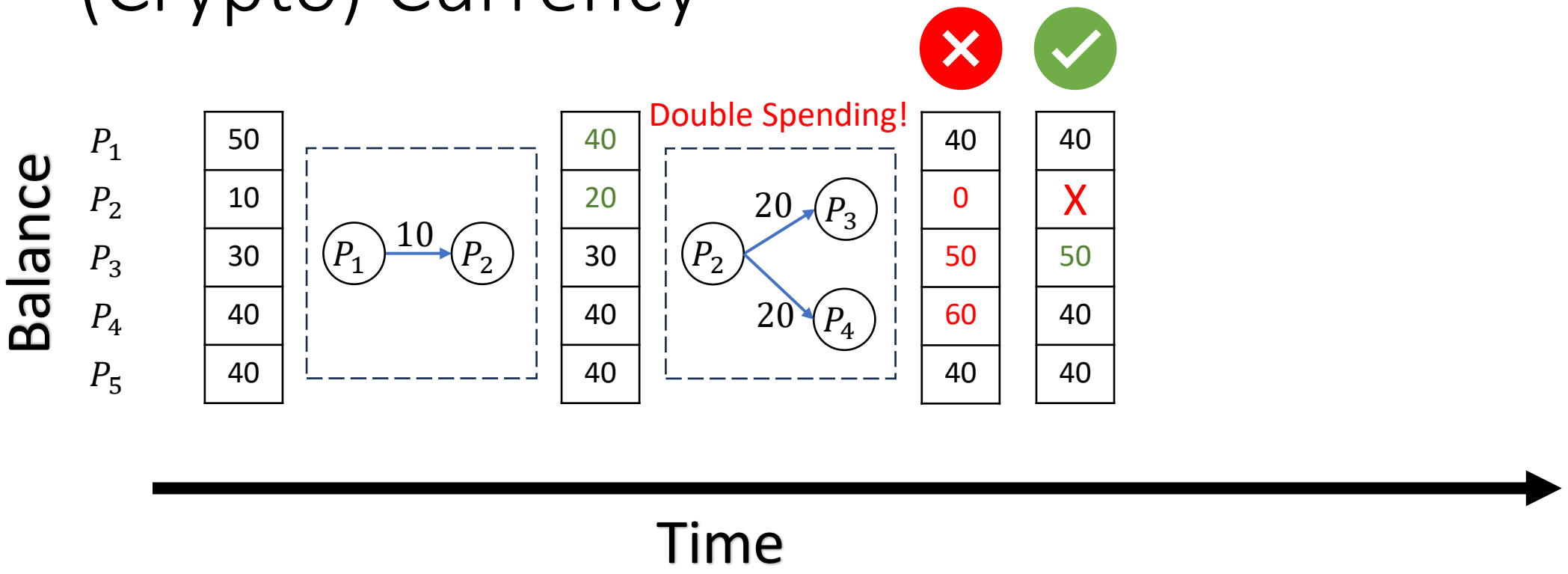
Trusting Randomness with Money

Gambling



Security

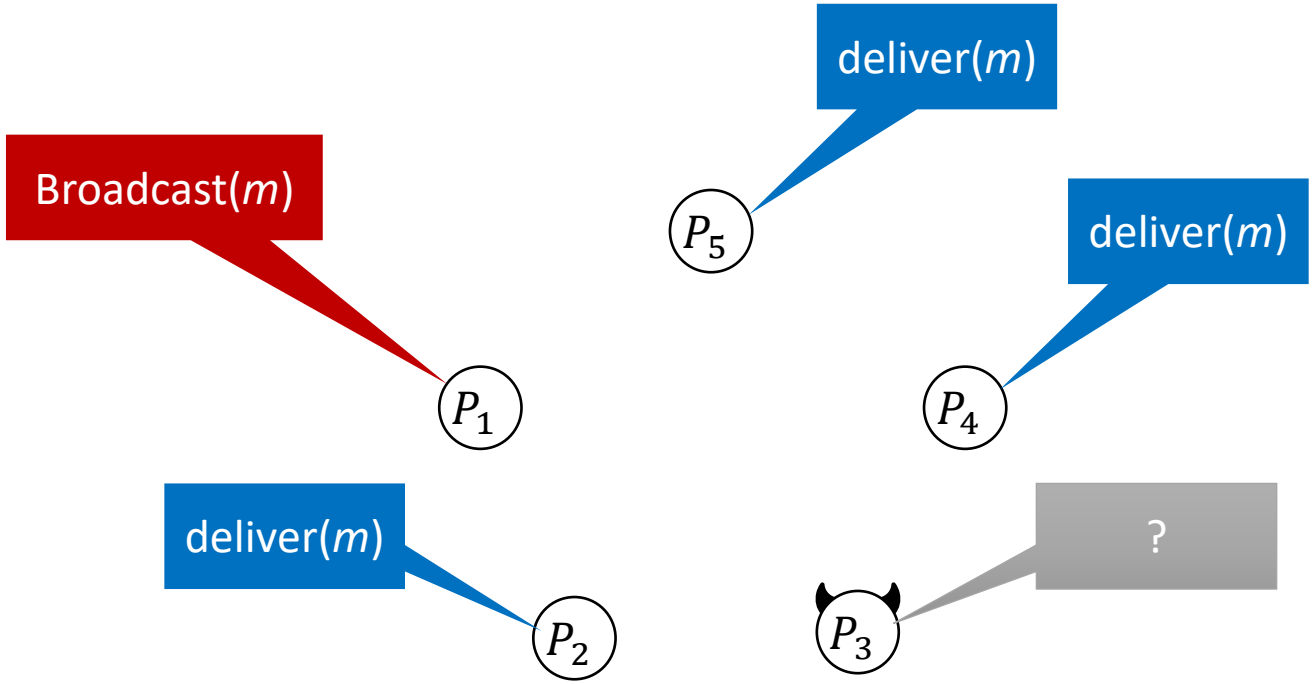
(Crypto) Currency



How to prevent double spendings?

- Shared view of everyone's balance
  - Centralized DB (bank)
  - Replicated state (e.g., blockchain)
- Consistent views of everyone's balance
  - Reliable broadcast [Guerraoui et al. 2019]

Reliable Broadcast (multi-instance)



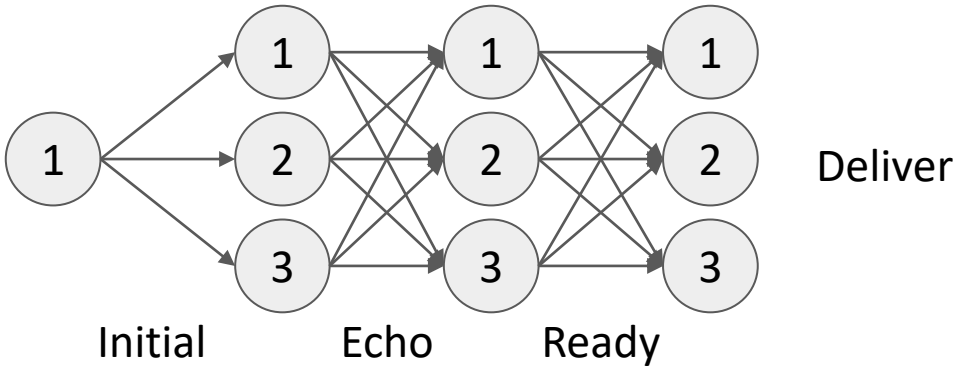
$$m = (\underbrace{\langle origin \rangle, \langle tag \rangle}_{\langle instance \rangle}, \langle value \rangle)$$

Reliable Broadcast (multi-instance)

- Integrity
  - **If** non-faulty Q delivers m with non-faulty origin P **then** P called broadcast(m).
- Validity
  - **If** non-faulty P broadcast m, **then** all non-faulty will deliver m
- Consistency
  - **If** non-faulty Q and R deliver m=(P,t,v) and m`=(P,t,v`) **then** v=v` (even if P is faulty)
- Totality
  - **If** non-faulty Q receives m **then** any non-faulty R will receive m

Reliable Broadcast cont.

- Available when #faulty < n/3
- Message complexity $\Theta(n^2)$ [Dolev et al 1985] [Bracha 1987]

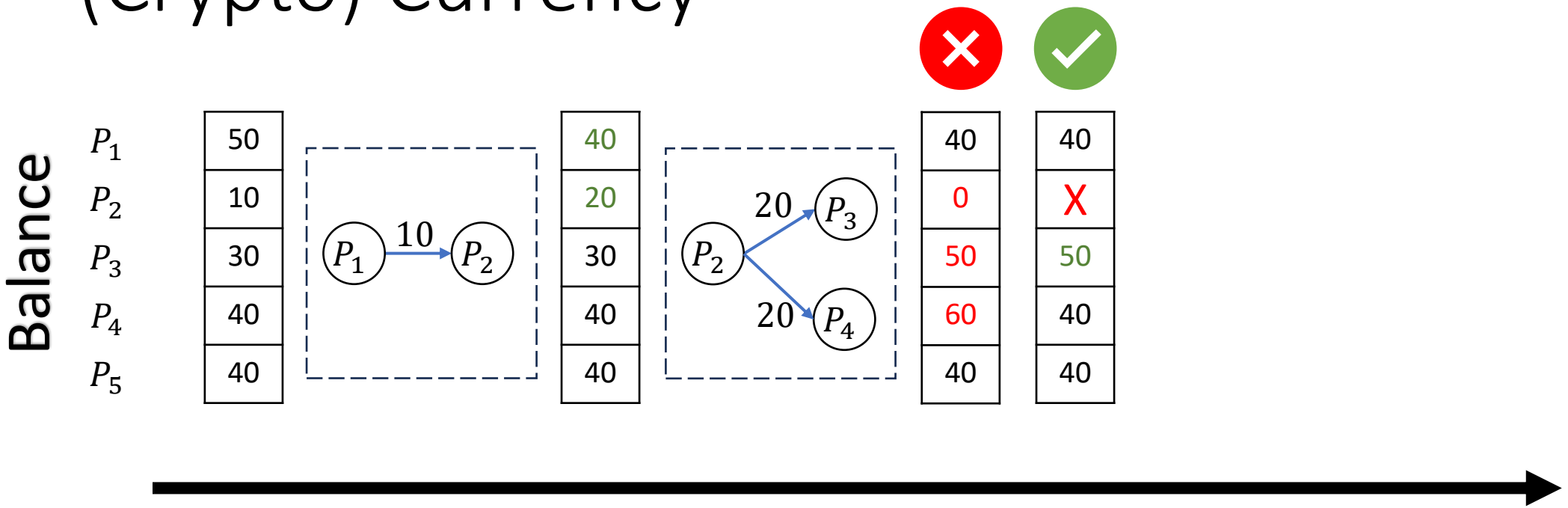


Bracha G. Asynchronous Byzantine Agreement Protocols, INFORMATION AND COMPUTATION 75, 130-143 (1987)

Reliable Broadcast cont.

- Available when #faulty < n/3
- Message complexity $\Theta(n^2)$ [Dolev et al 1985] [Bracha 1987]
- Can prevent double spending!
  - by forcing per sender ordering

(Crypto) Currency



$$m = (P_2, 1, P_2 \xrightarrow{20} P_3)$$

$$m' = (P_2, 1, P_2 \xrightarrow{20} P_4)$$

Reliable Broadcast cont.

- Available when #faulty < n/3
- Message complexity $\Theta(n^2)$ [Dolev et al 1985] [Bracha 1987]
- Can prevent double spending!
  - by forcing per sender ordering
  - Only one message per process and index will be received

$$m = (P_2, 1, P_2 \overset{20}{\rightarrow} P_3)$$

$$m' = (P_2, 1, P_2 \overset{20}{\rightarrow} P_4)$$

Reliable Broadcast cont.

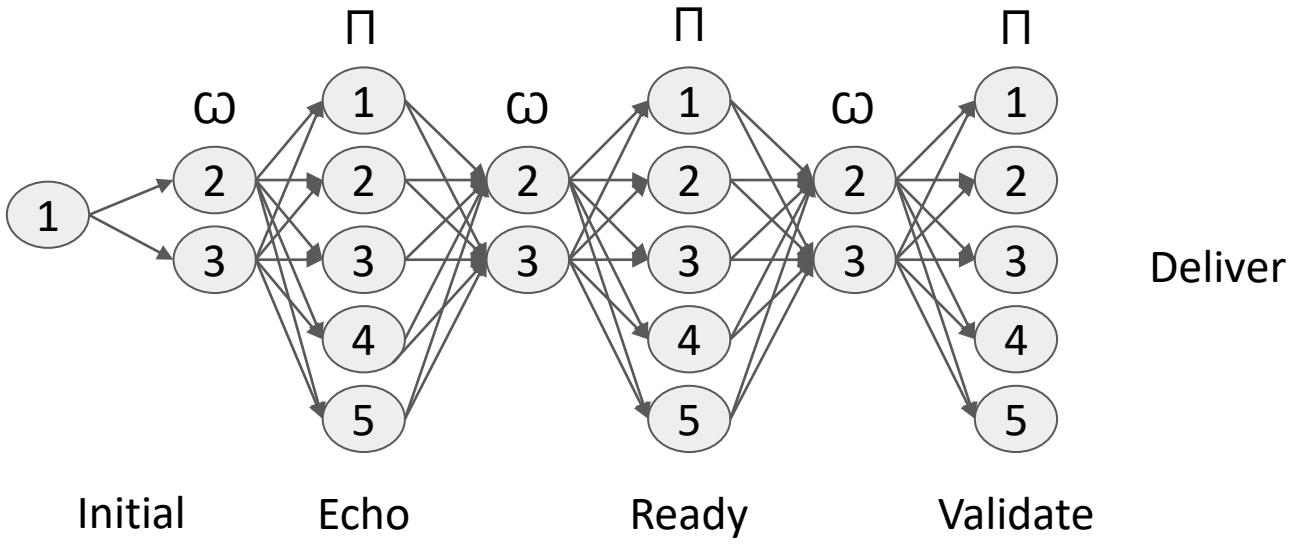
- Available when #faulty < n/3
- Message complexity $\Theta(n^2)$
- Can prevent double spending!

Can we improve it?

What if we had a trusted set?

Reliable broadcast with trusted witness sets (quorums)

- Each message is sent and/or verified by the witness set
- Message complexity: $\Theta(w \cdot n)$ where w is the witness set size.



This work (extended protocol)

Reliable broadcast with trusted witness sets (quorums)

- Each message is sent and/or verified by the witness set
- Message complexity: $\Theta(w \cdot n)$ where w is the witness set size.
- Challenges:
  - How to select such witness sets?
  - How to prevent an adaptive attacker from compromising them?

Idea: select a random set per message/transaction

Random selection of witness sets

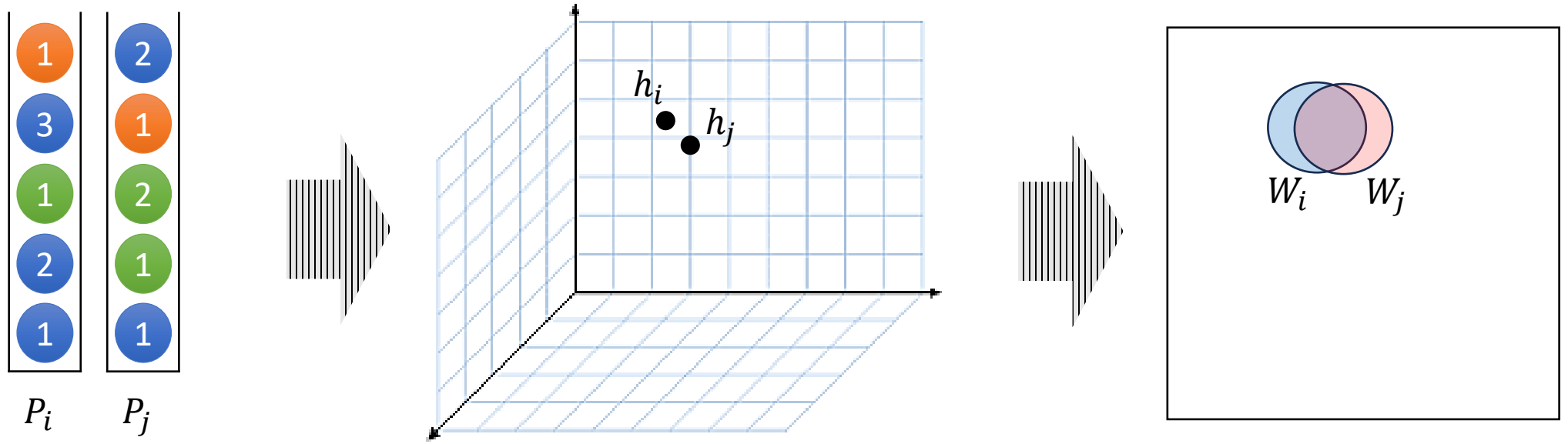
- Benefits
  - The witness set has similar ratio of faulty members like in the network W.H.P
  - Unpredictable and therefore hard to compromise in advance

- Existing approaches
  - MPC of shared randomness – high complexity
  - Use previous blockchain block hash [Algorand] – requires synchronous blocks

- **Goal: Random selection for high-rate transactions**

Our approach: locality sensitive history hash

- Independently computed by each process (no communication latency)
- Similar histories are hashed to similar values (weak synchronization)
- Similar hash values results with similar (heterogenous) sets



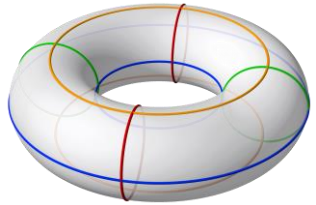
Process histories

Hash space

Processes

Our approach: locality sensitive history hash

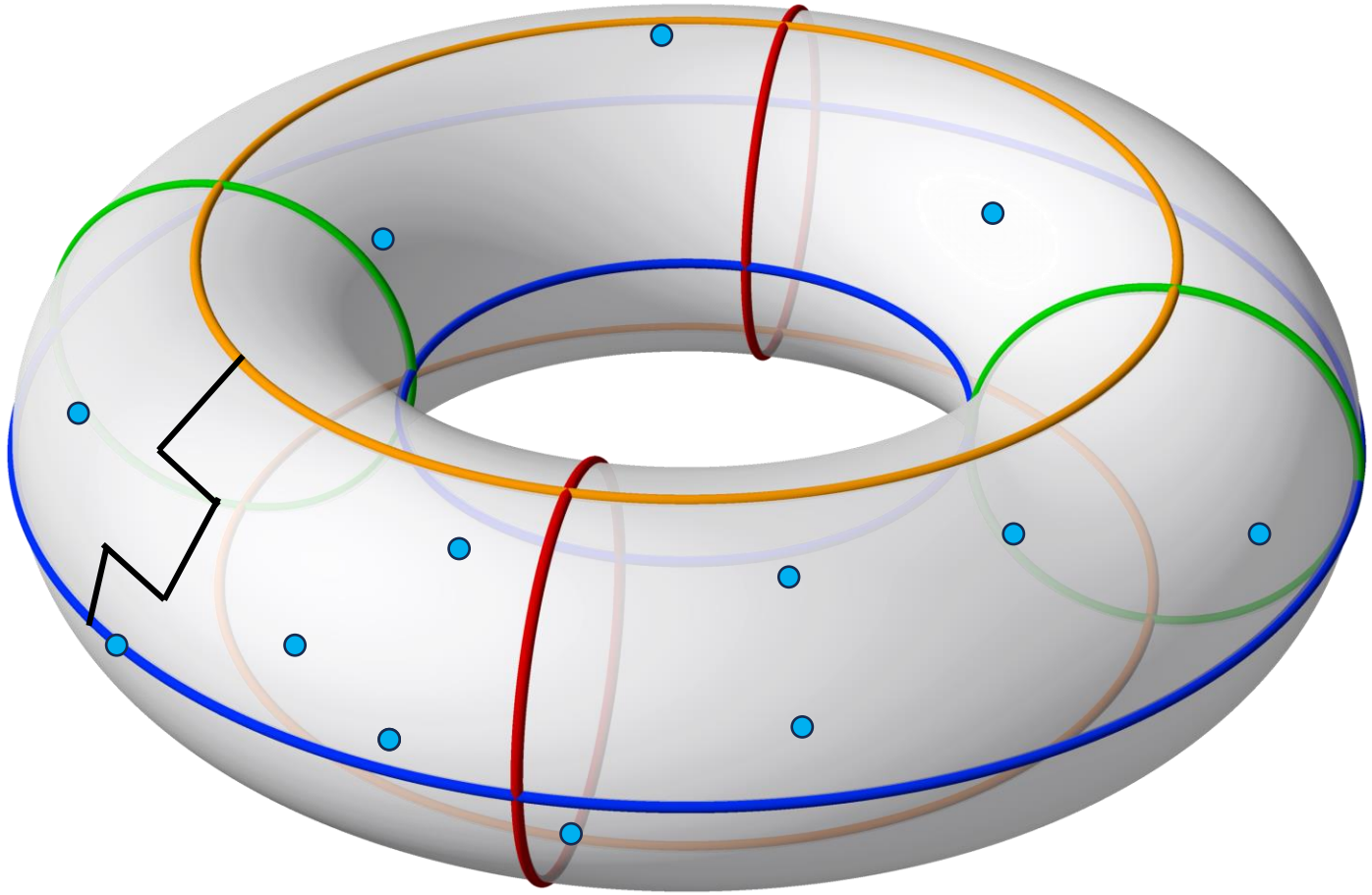
- Challenges
  - Histories my differ by some recent messages and the total ordering
  - Hash should be hard to predict
  - Quickly converge to uniform (compared to attack time)



- Solution
  - History is hashed to a vector ***v*** in a cyclic D dimensional space $\mathbb{Z}_r^D$ (torus)
  - Each message is hashed to dimension $d \in [D]$ and a direction $t \in \{-1, 1\}$, and used to update ***v[d] += t***
  - Peer IDs are also hashed to the same hash space
  - Peers are selected based on proximity to the history hash (at a given time)

Random walk in a torus

- History advances like a random walk in a torus.
- Peers are uniformly scattered.

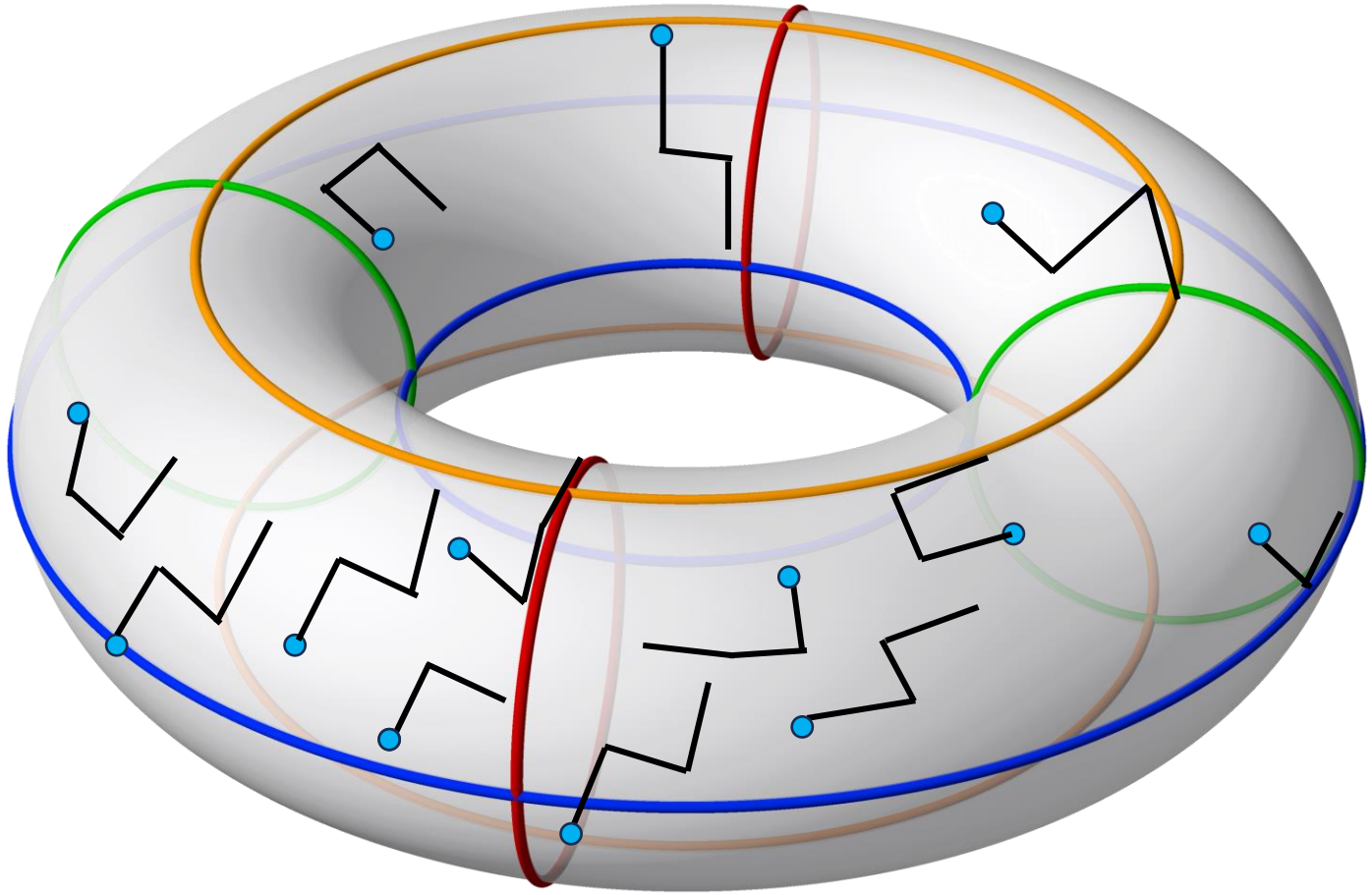


Random walk in a torus

- History advances like a random walk in a torus.
- Peers are uniformly scattered.

- Extension:

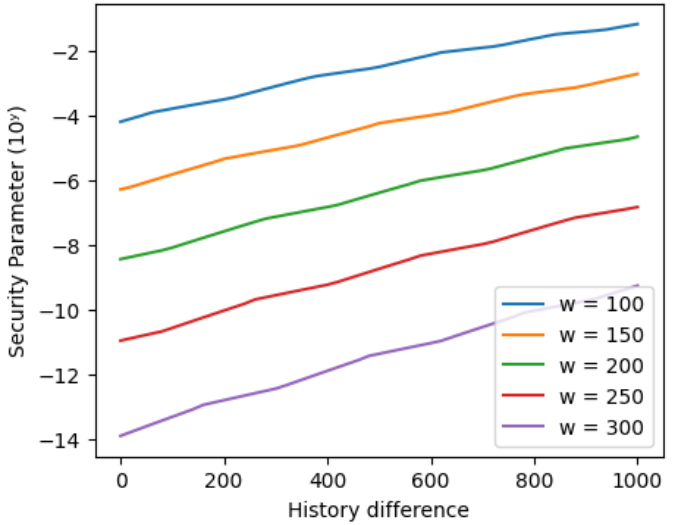
  History hash is used to derive independent per peer and transaction random walks



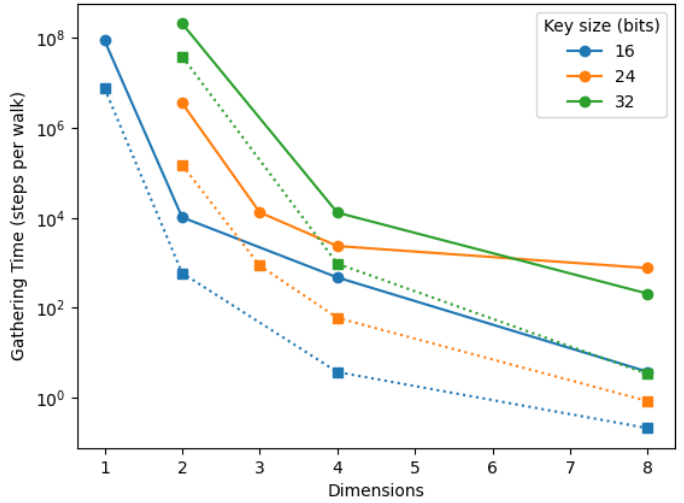
Technical Report: “Dynamic Probabilistic Reliable Broadcast” [arxiv.org/abs/2306.04221]

Can we trust these witness sets?

- Reliable broadcast requirements are provided W.H.P
- Probability that a witness set is comprised is very low
  - $\Pr(faulty(W) < k) = \sum_{i=0}^{k-1} \binom{f}{i} p^i (1-p)^{f-i}$

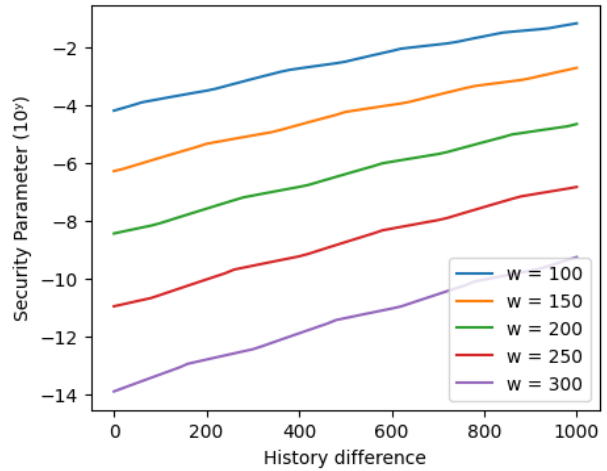


- Time till many compromised peers are selected together is very large
  - (Simulation + approximation)



Can we use external sources of trust?

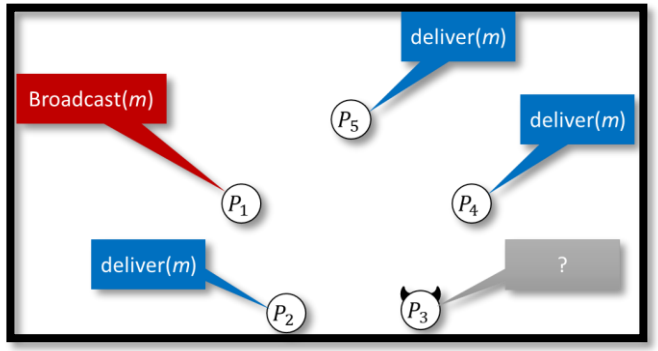
- Adaptive witness set size based on message issuer (float trust)
  - Trust can be defined by $\{p_i\}_{i\in[n]}$
    - where p_i = probability that i will try double spending
  - We can keep a constant failure probability
    - $p_{fail} = p_i \cdot Pr(\#faulty(W_i) > k)$
  - Faster transactions within trust cliques

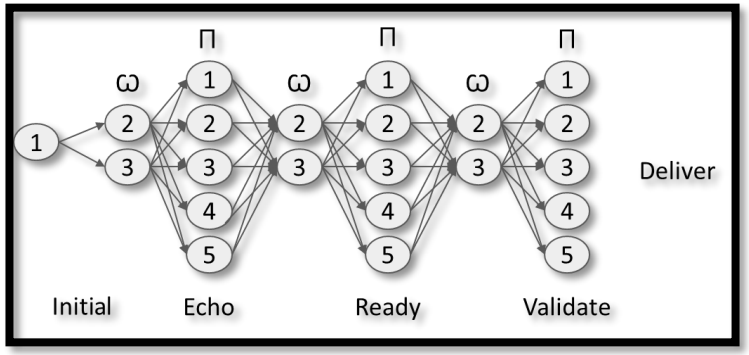


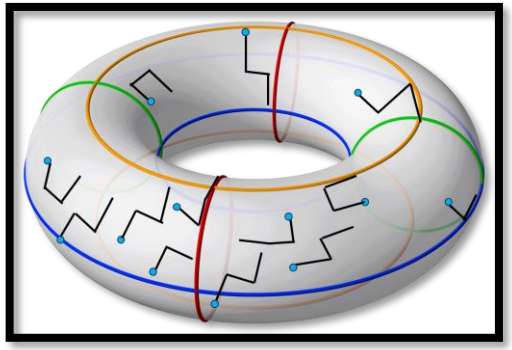
- Extending witnesses with trusted peers (Boolean trust)
  - Considering
    - logarithmic size witness set ($\mathrm{w} = \mathrm{c}\ln n$)
    - Constant trust ratio T (each peer trusts $\mathrm{T}\cdot n$ peers)
    - Each peer uses the random set but if needed it also uses an extra (closest) trusted peer
  - Most of the times the random set can be trusted as is (constant number of extra peers)
  - Whenever $T \geq 1 - e^{-\frac{1}{c}}$, for example, if $c = 1$ and $T > \frac{2}{3}$ or $c = 10$ and $T > \frac{1}{10}$

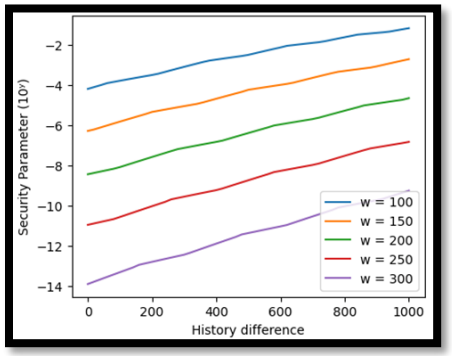
Summary

- Crypto currency using reliable broadcast
- Low-cost reliable broadcast using trusted witness sets
- Random selection of witnesses using local sensitive hashing of history
- Using external sources of trust









Thank you

In memory of Israeli men, women and children slaughtered by Hamas and wishing for the release of all those abducted and still held in Gaza.