Efficient and Secure Distributed Consensus

Charalampos Papamanthou
Shravan Srinivasan, Georgios Tsimos

https://eprint.iacr.org/2020/894

Blockchains, MPC protocols, and more require Distributed Consensus. Need to:

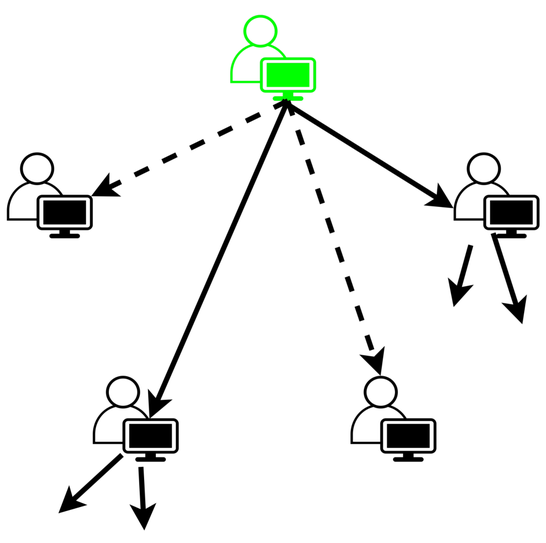
Improve efficiency in dishonest majority:

- **Communication Complexity**
- **Round Complexity**

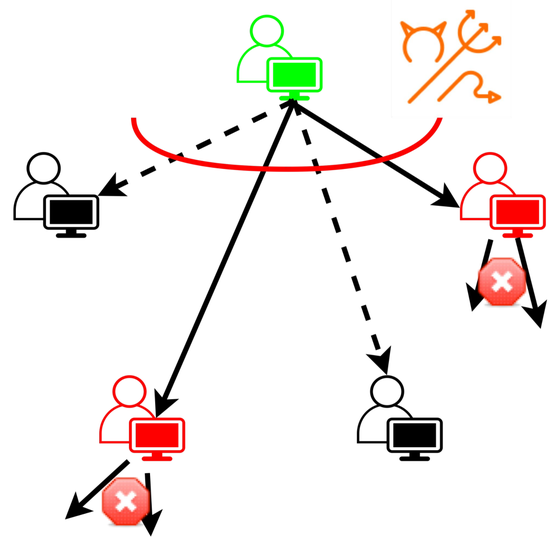
Improve security to tolerate network adversaries:

- **Strongly adaptive** (delete messages *in-flight*)
- Mobile sluggish faults (weakly synchronous)

Gossiping for Communication Efficiency



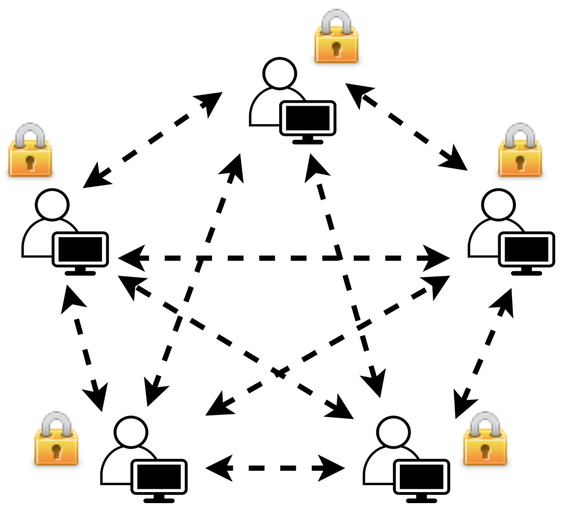
Gossiping: Send to random parties. Guaranteed propagation after a few gossiping rounds.



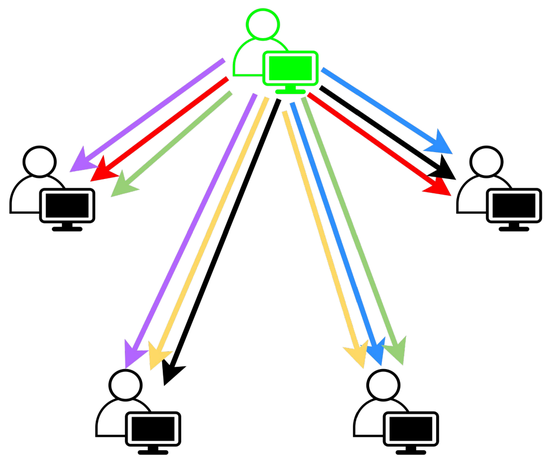
Attack: Adversary observes message pattern and corrupts receivers.

Communication Efficiency:

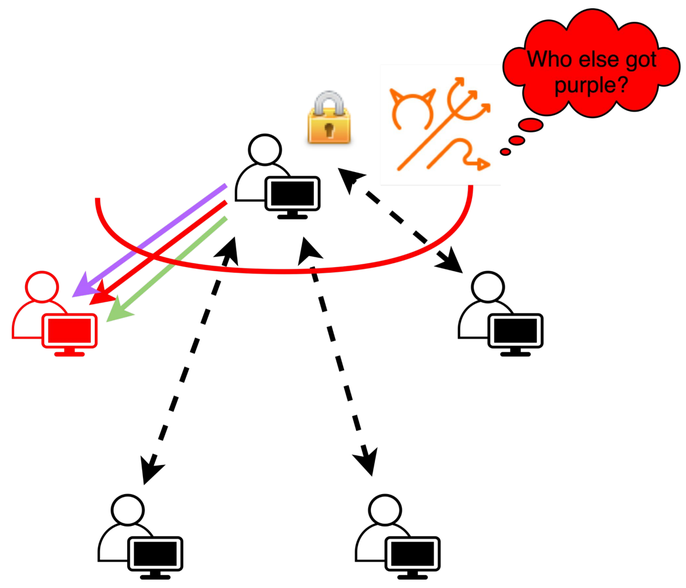
- Improved SoA by **O(n)** in (P)BC **without** trust assumptions
- Improved SoA by **O(n)** in PBC assuming trusted PKI
- New propagation primitive vs. weakly adaptive adversaries



PBC: More messages allow for efficient amortized communication.

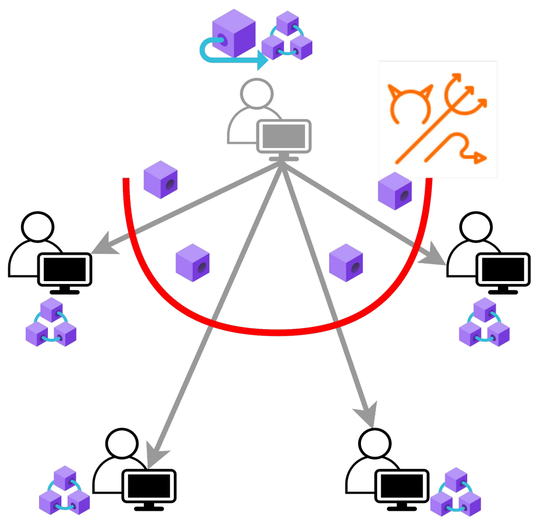


Propagation: Combine message gossiping: send similar-sized, random lists of messages.

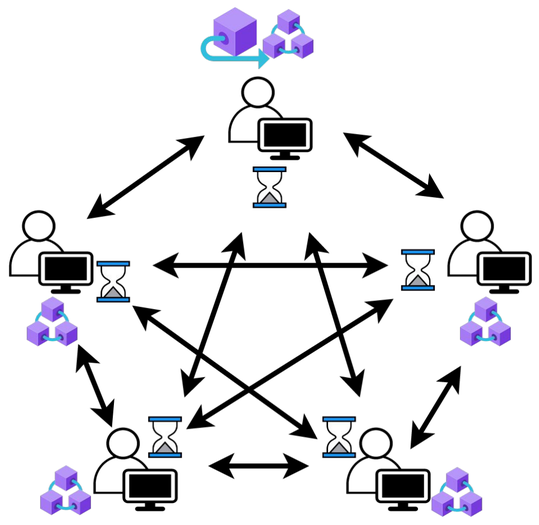


Security: Adversary cannot distinguish between encrypted lists received by honest parties.

Tolerate network adversaries



Attack: Adversary can learn the contents before deleting or delaying a message.



Security: Block winner time-lock encrypts. Others send a decoy.

Security:

- Time-lock puzzle are building blocks
- Mobile sluggish faults in Nakamoto
- Compiler to convert weakly to strongly adaptive
- Expected **O(1)** BC (in strongly adaptive & dishonest majority)


Time-lock Puzzle


Blockchain


Encrypted Message


Adversary