Tardigrade:

An atomic broadcast protocol for arbitrary network conditions

Erica Blum
University of Maryland

Jonathan Katz
University of Maryland

Julian Loss
CISPA Hemholtz Center for Information Security

What is atomic broadcast?

Atomic broadcast:

- Parties receive input values over time
- Want to agree on a growing, ordered *sequence* of values
- Some parties are **Byzantine** (faulty)

Our setting:

- Point-to-point authenticated channels
- Messages sent by an honest party eventually arrive (i.e., messages cannot be dropped)
- Assume trusted dealer performs setup for public key infrastructure (PKI), threshold signatures, threshold encryption

Network models

Synchronous model

- Messages arrive within fixed delay $\leq \Delta$
- Optimal fault tolerance*: $t < n/2$

Asynchronous model

- No upper bound on message delay
- Optimal fault tolerance*: $t < n/3$

*assuming PKI

How do I know which one to choose?
What happens if I'm wrong?

	Delay $\leq \Delta$, $t < n/2$	Unbounded delay, $t < n/3$
Async. Protocol	**Not secure for $t \geq n/3$ faults**	Secure
Sync. Protocol	Secure	**Not secure if delay exceeds Δ**

Prior work

Other settings:

- Network may be synchronous or partially synchronous [MNR19, MR21]
- Temporary partitions/“sluggish” faults [e.g. GPS17, AMNRY20]
- Tolerating $t < n/2$ *crash* faults if network is asynchronous [LVCQV16]

Most closely related:

- Network-agnostic Byzantine agreement [BKL19] and secure multiparty computation [BLL20]

Research question

Can we design a “network-agnostic” protocol that is

- Secure for $t_s > n/3$ faults if network is synchronous
- Secure for $0 \leq t_a \leq t_s$ faults otherwise?

Our contributions

1. **Lower bound**: if $t_a + 2t_s \geq n$, then there is no atomic broadcast protocol secure against t_s faults in a sync. network and t_a faults in an async. network.

2. **Tardigrade**: a network-agnostic atomic broadcast protocol
   - Optimal fault tolerance (any t_a, t_s such that $t_a + 2t_s < n$)
   - Can be made adaptively secure

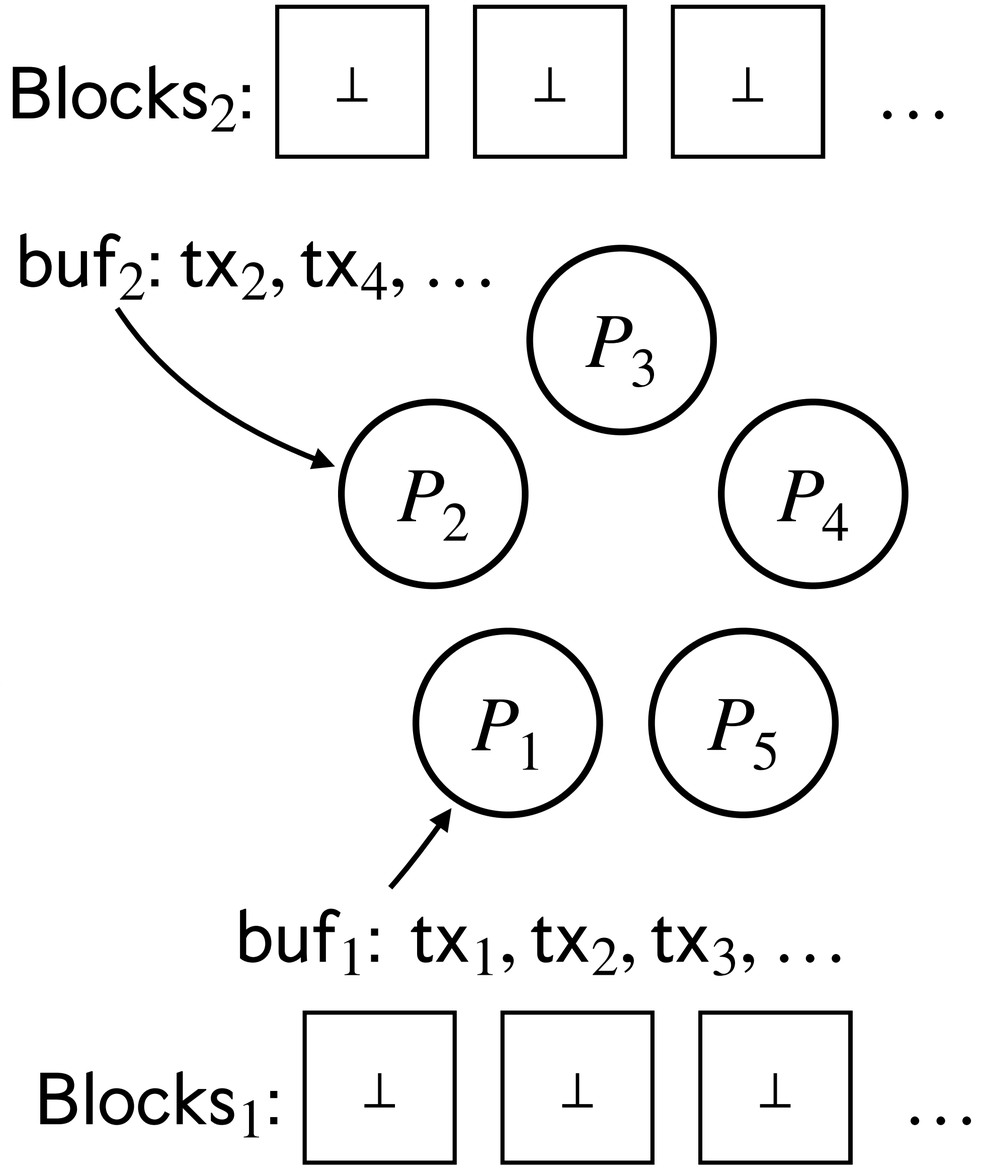
3. **Upgrade**: network-agnostic atomic broadcast with asymptotic communication complexity matching state of the art async. protocols, at the cost of tolerating $O(\epsilon)$ fraction fewer faults. (Only statically secure)

Atomic broadcast

- Each party P_i has a local buffer buf_i and array of blocks Blocks_i
- Values are added to buffers over time

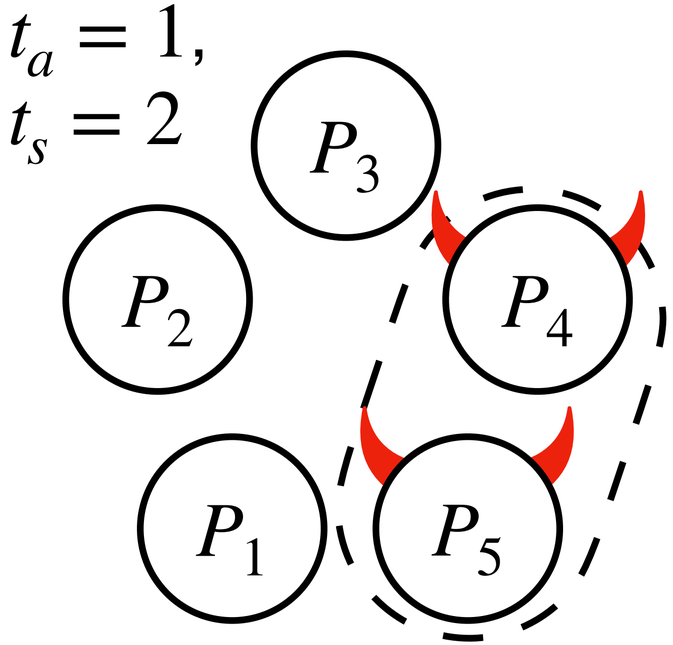
Security properties:

- <u>Consistency:</u> if P_i, P_j have both output block k, then $\mathsf{Blocks}_i[k] = \mathsf{Blocks}_j[k]$
- <u>Completeness:</u> each party eventually outputs a block at index k (for all k)
- <u>Liveness:</u> if tx is in all honest parties' buffers, then each party eventually outputs a block that contains tx



Technical overview: lower bound

Theorem: There is no atomic broadcast protocol secure against t_s faults in a sync. network and t_a faults in an async. network for $t_a + 2t_s \geq n$.



Honest P_1 has not heard from P_4, P_5.

Explanation 1: Network is asynchronous and P_4, P_5 are honest

Explanation 2: Network is synchronous and P_4, P_5 are malicious

Interlude: What is a tardigrade?

Milnesium tardigradum

- Type of microscopic animal also known as “water bears”
- Capable of surviving extreme heat, cold, radiation, and pressure by entering a state called *cryptobiosis*
- In one experiment, 68% of tardigrades survived exposure to hard vacuum of outer space

Technical overview: Tardigrade

Share inputs:

Send (buf_i, σ_i) to all

Wait to collect "pre-block" $B_i = \{(\text{buf}_j, \sigma_j)\}_{j \in P' \subseteq P}$ containing $n - t_s$ inputs

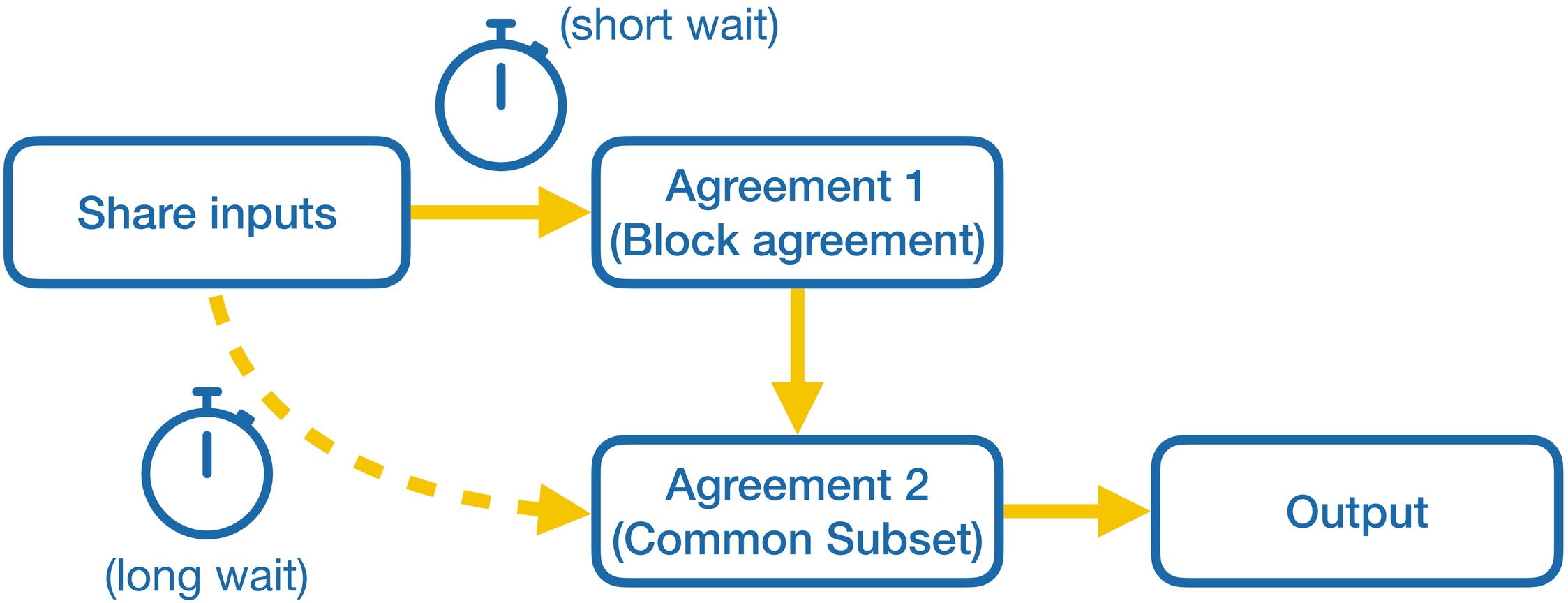
Agreement 1&2:

Agree on set of pre-blocks

Output:

Output new block

Technical overview: Tardigrade



Technical overview: Tardigrade

Why it works

	Sync. network (t_s faults)	Async. network (t_a faults)
Agreement 1 (Block agreement)	**Terminates in time with consistent output**	If it terminates, then the output is valid
Agreement 2 (Common Subset)	If all honest parties input S, then all honest parties terminate with output S	**Terminates eventually with consistent output**

Recap

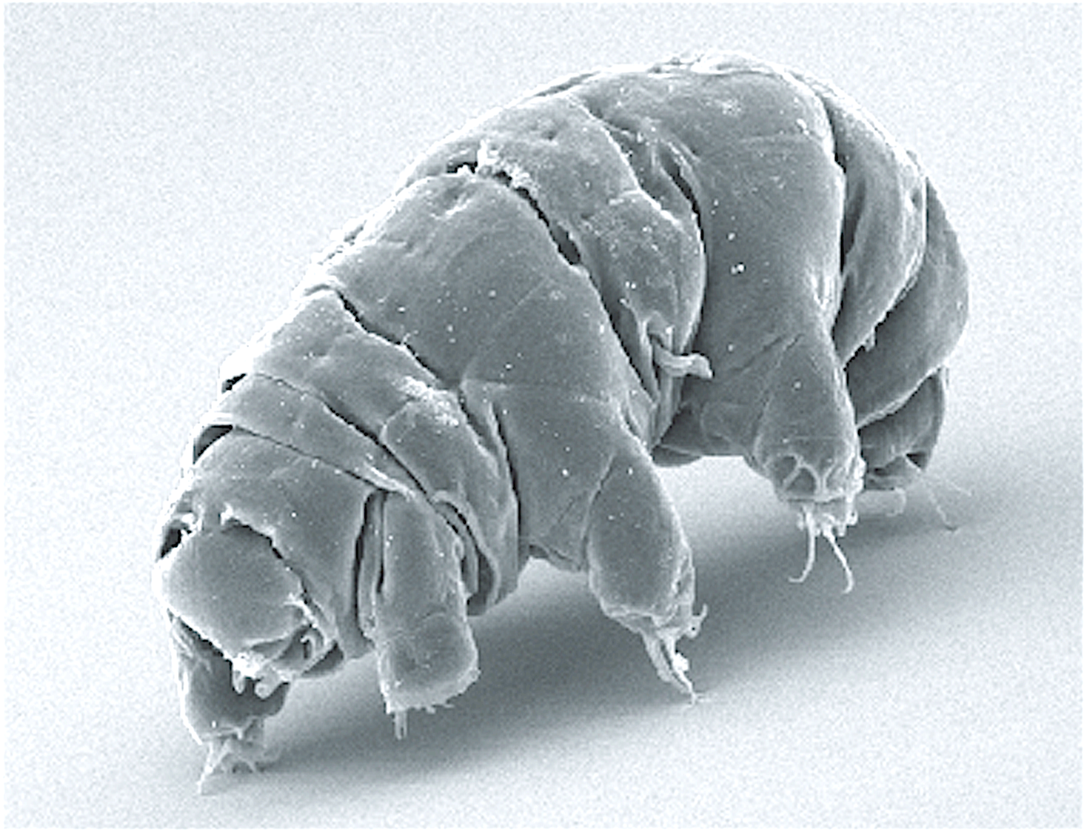


Image attribution: Schokraie E, Warnken U, Hotz-Wagenblatt A, Grohme MA, Hengherr S, et al. (2012) / Wikimedia Commons / CC BY 2.5

Tardigrade facts: https://en.wikipedia.org/wiki/Tardigrade

Lower bound: if $t_a + 2t_s \geq n$, there is no atomic broadcast protocol secure against t_s faults in a sync. network and t_a faults in an async. network.

Tardigrade: network-agnostic atomic broadcast with optimal fault tolerance

Upgrade: better communication complexity for $O(\epsilon)$ fewer faults

Full paper: eprint.iacr.org/2020/142.pdf

References

E. Blum, J. Katz, and J. Loss. Synchronous consensus with optimal asynchronous fallback guarantees. In *17th Theory of Cryptography Conference—TCC 2019*. Springer, 2019.

E. Blum, C.-D. L. Zhang, and J. Loss. Always have a backup plan: Fully secure synchronous MPC with asynchronous fallback. In *Advances in Cryptology— Crypto 2020*. Springer, 2020.

I. Abraham, D. Malkhi, K. Nayak, L. Ren, and M. Yin. Sync HotStuff: Simple and practical synchronous state machine replication. In *2020 IEEE Symposium on Security and Privacy*. IEEE, 2020.

Y. Guo, R. Pass, and E. Shi. Synchronous, with a chance of partition tolerance. *In Advances in Cryptology —Crypto 2019.* Springer, 2019.

S. Liu, P. Viotti, C. Cachin, V. Quema, and M. Vukolic. XFT: Practical fault tolerance beyond crashes. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, USENIX, 2016.

D. Malkhi, K. Nayak, and L. Ren. Flexible byzantine fault tolerance. In *26th ACM Conf. on Computer and Communications Security (CCS).* ACM Press, 2019.

A. Momose and L. Ren. Multi-threshold Byzantine fault tolerance. In *28th Conference on Computer and Communications Security (CCS),* 2021. Available at https://eprint.iacr.org/2017/307.