Asynchronous Byzantine Agreement with Subquadratic Communication

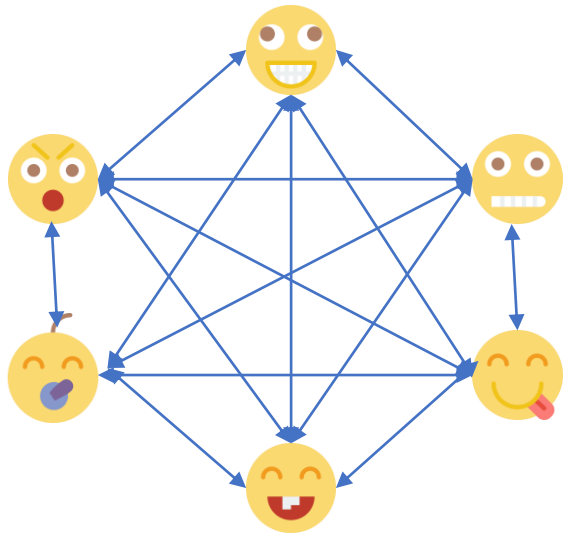
Erica Blum — U. Maryland

Jonathan Katz — U. Maryland

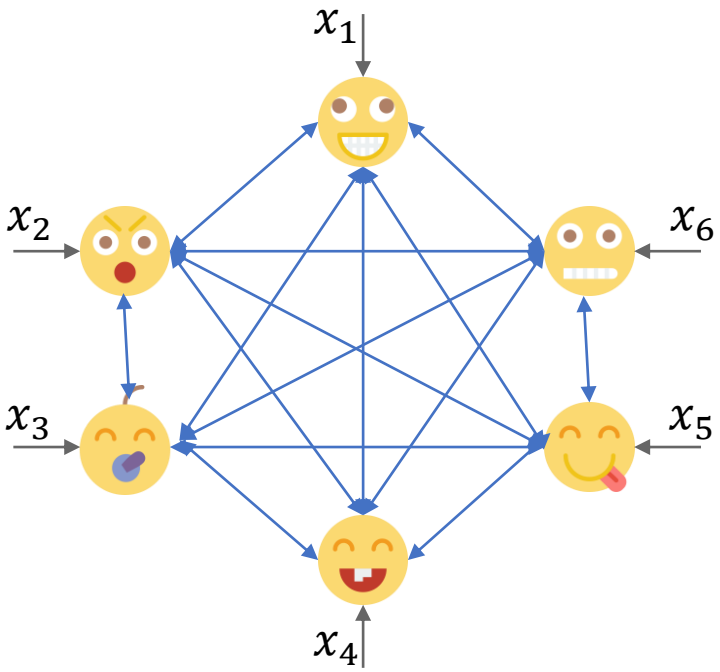
Chen-Da Liu-Zhang — ETH Zurich

Julian Loss — U. Maryland

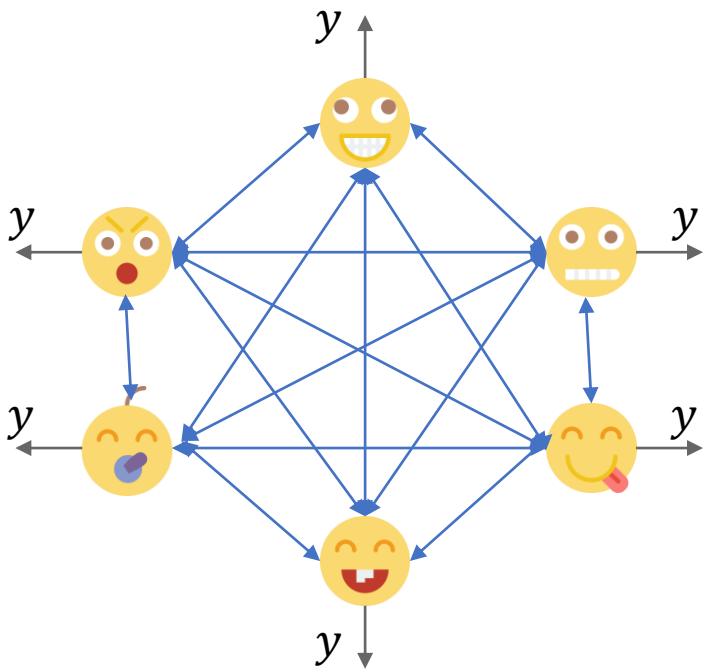
Byzantine Agreement



Byzantine Agreement

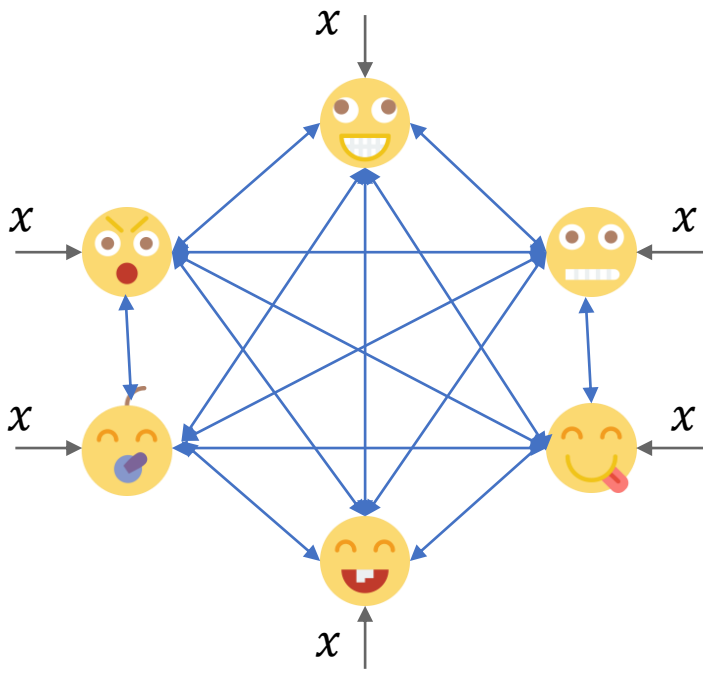


Byzantine Agreement



All honest parties agree on the same output

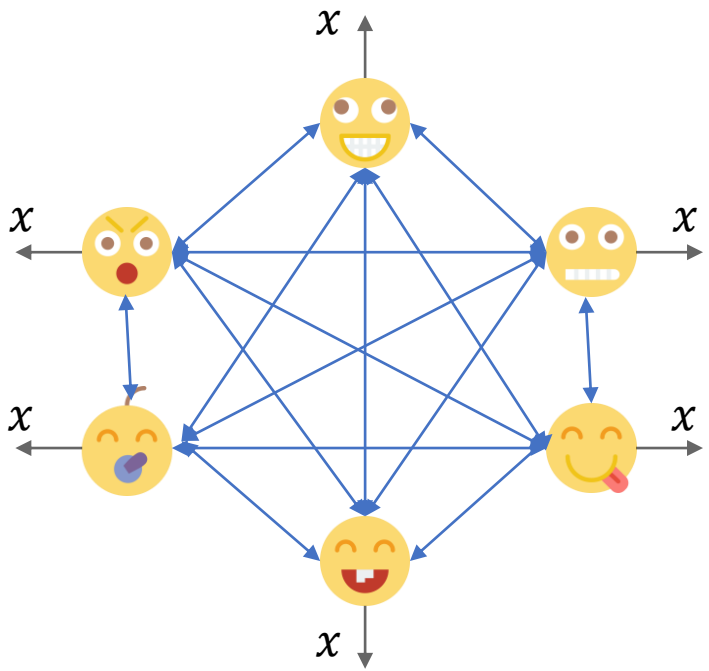
Byzantine Agreement



All honest parties agree on the same output

If honest parties have the same input, they keep the same value as output

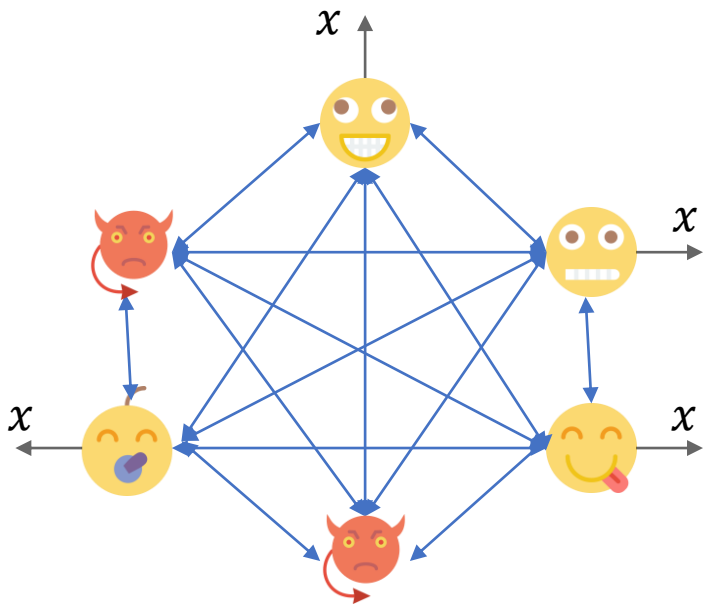
Byzantine Agreement



All honest parties agree on the same output

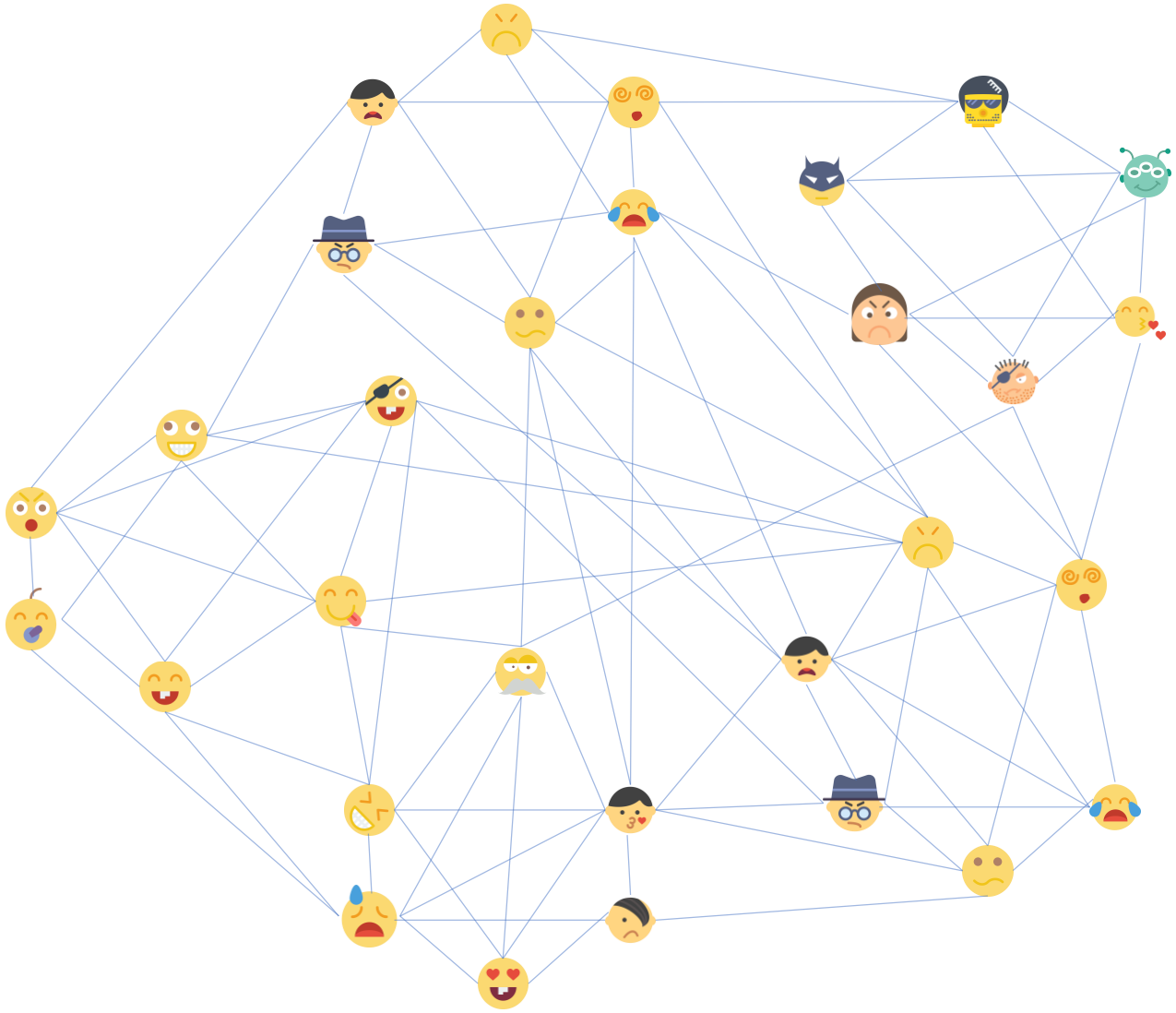
If honest parties have the same input, they keep the same value as output

Byzantine Agreement



All honest parties agree on the same output

If honest parties have the same input, they keep the same value as output



Is there an asynchronous BA with $o(n^2)$ communication that tolerates $\theta(n)$ adaptive corruptions?

Is there an asynchronous BA with $o(n^2)$ communication that tolerates $\theta(n)$ adaptive corruptions?

- Feasibility of asynch. $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ using a trusted dealer (alternately, with *amortized* $o(n^2)$ and without setup)

Is there an asynchronous BA with $o(n^2)$ communication that tolerates $\theta(n)$ adaptive corruptions?

- Feasibility of asynch. $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ using a trusted dealer (alternately, with *amortized* $o(n^2)$ and without setup)

- Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ corruptions without setup

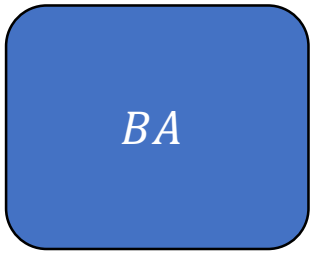
Related Work

Most previous subquadratic BA are synchronous or partially synchronous [KS06,KS10,M17,A+19,…]

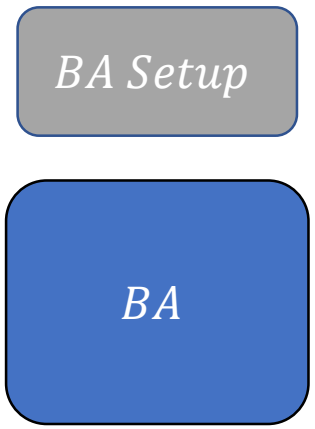
Recent work by Cohen et al. [CKS20] give subquadratic asynchronous BA, but the adversary has restricted scheduling power

Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive

Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



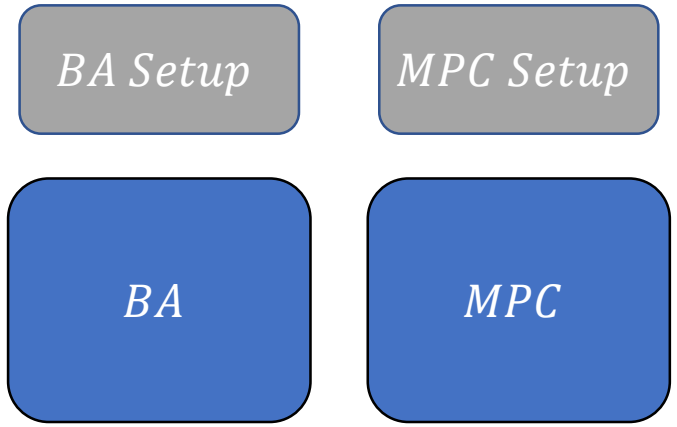
Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



Size: $O\big(poly(\kappa)\big)$

CC: $O(poly(\kappa) \cdot n)$

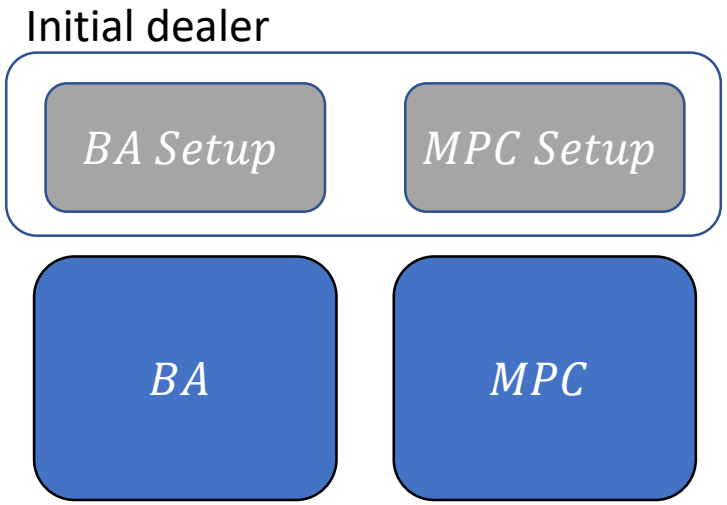
Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



Size: $O\big(poly(\kappa)\big)$

CC: $O(poly(\kappa) \cdot n)$

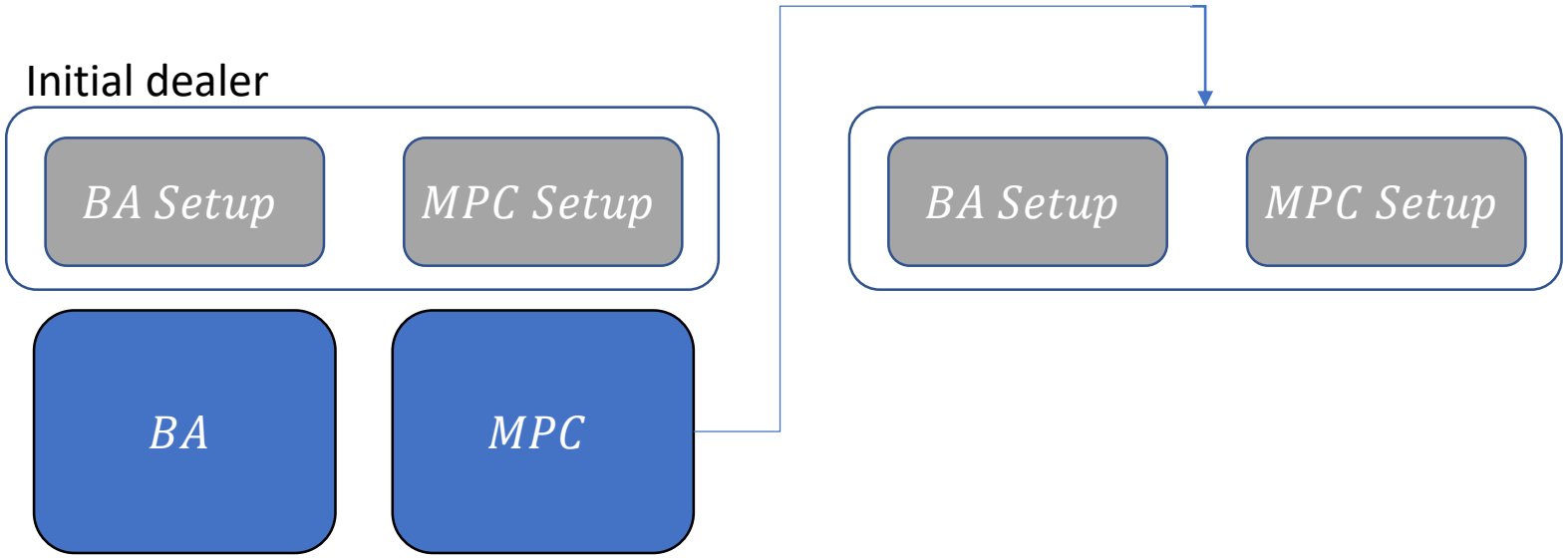
Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



Size: $O\big(poly(\kappa)\big)$

CC: $O(poly(\kappa) \cdot n)$

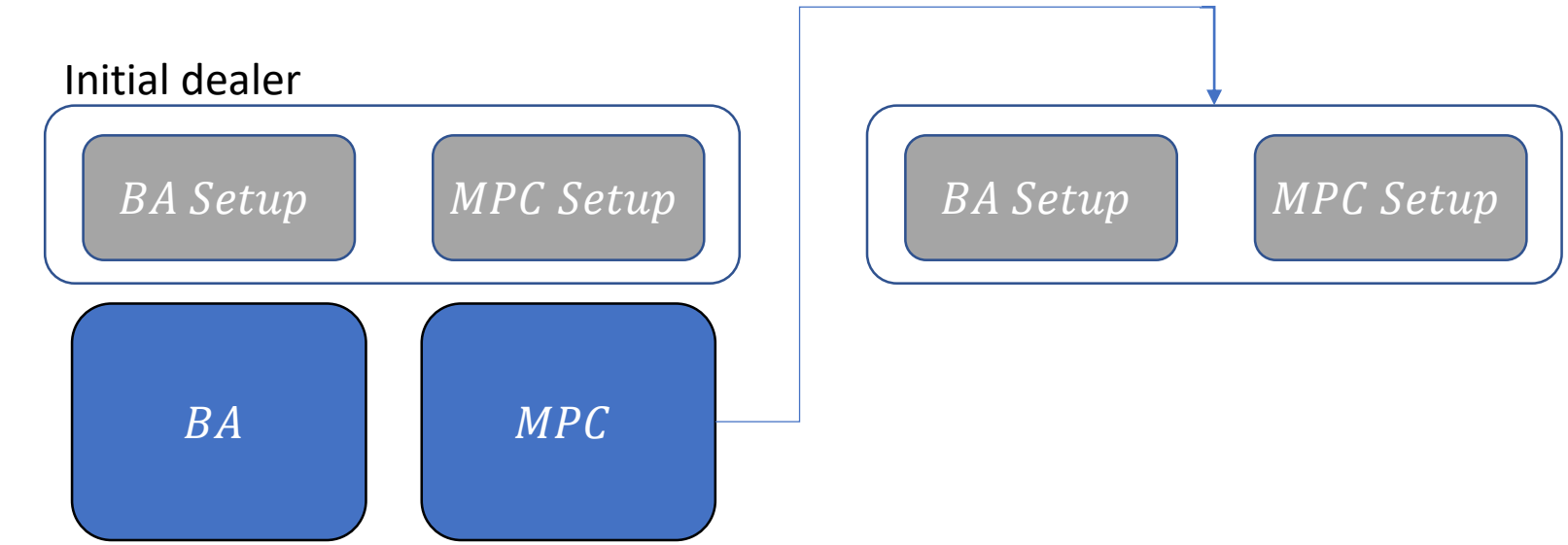
Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



Size: $O\big(poly(\kappa)\big)$

CC: $O(poly(\kappa) \cdot n)$

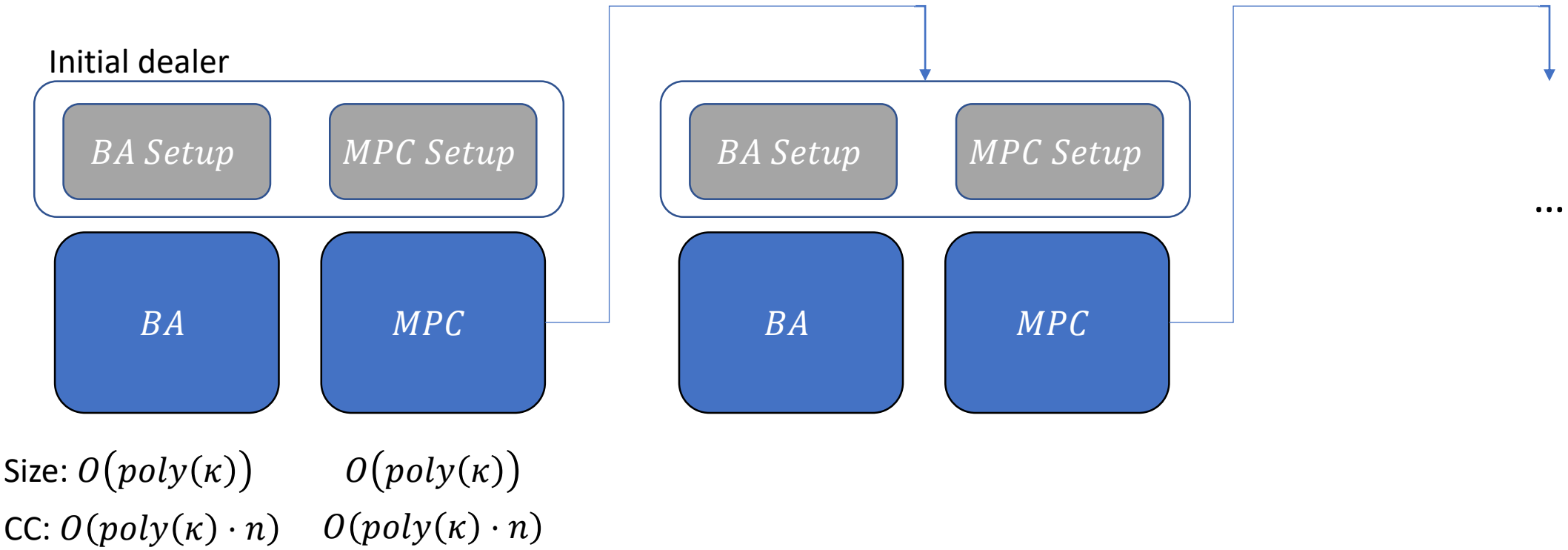
Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



Size: $O(poly(\kappa))$ $O(poly(\kappa))$

CC: $O(poly(\kappa) \cdot n)$ $O(poly(\kappa) \cdot n)$

Feasibility of asynchronous $o(n^2)$ BA for $f < (1-\epsilon)\,{}^{n}/_{3}$ adaptive



One-Time BA

One-Time BA

Graded Consensus [CR93]
Input x_i; Output (z_i, g_i)

If ∀ honest P_i $x_i = x$, then $(z_i, g_i) = (x, 1)$
If ∃ honest P_i $g_i = 1$, then $z_j = z_i$

One-Time BA

Graded Consensus [CR93]
Input x_i; Output (z_i, g_i)

If ∀ honest P_i $x_i = x$, then $(z_i, g_i) = (x, 1)$
If ∃ honest P_i $g_i = 1$, then $z_j = z_i$

Coin-Flip
Each P_i obtains the same random bit c_i

One-Time BA

$\leq O(\kappa)$

Graded Consensus [CR93]
Input x_i; Output (z_i, g_i)

If ∀ honest P_i $x_i = x$, then $(z_i, g_i) = (x, 1)$
If ∃ honest P_i $g_i = 1$, then $z_j = z_i$

Coin-Flip
Each P_i obtains the same random bit c_i

If $g_i = 0$: $x_i = c_i$
Else $x_i = z_i$

One-Time BA

$\leq O(\kappa)$

GC

Graded Consensus [CR93]
Input x_i; Output (z_i, g_i)

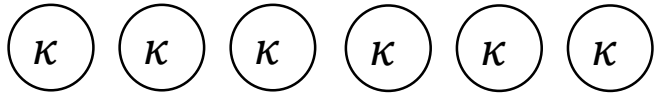
If $\forall$ honest P_i $x_i = x$, then $(z_i, g_i) = (x, 1)$
If $\exists$ honest P_i $g_i = 1$, then $z_j = z_i$

$Coin$

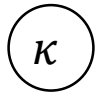
Coin-Flip
Each P_i obtains the same random bit c_i

If $g_i = 0$: $x_i = c_i$
Else $x_i = z_i$

$BA\ Setup$



Each party in set can prove membership



Each party in set has a (signed) share of c_i

One-Time BA

$\leq O(\kappa)$

GC

Graded Consensus [CR93]
Input x_i; Output (z_i, g_i)

If $\forall$ honest P_i $x_i = x$, then $(z_i, g_i) = (x, 1)$
If $\exists$ honest P_i $g_i = 1$, then $z_j = z_i$

$Coin$

Coin-Flip
Each P_i obtains the same random bit c_i

If $g_i = 0$: $x_i = c_i$
Else $x_i = z_i$

Communication $O(poly(\kappa) \cdot n)$
Setup size $O\big(poly(\kappa)\big)$

$BA\ Setup$

κ κ κ κ κ κ

Each party in set can prove membership

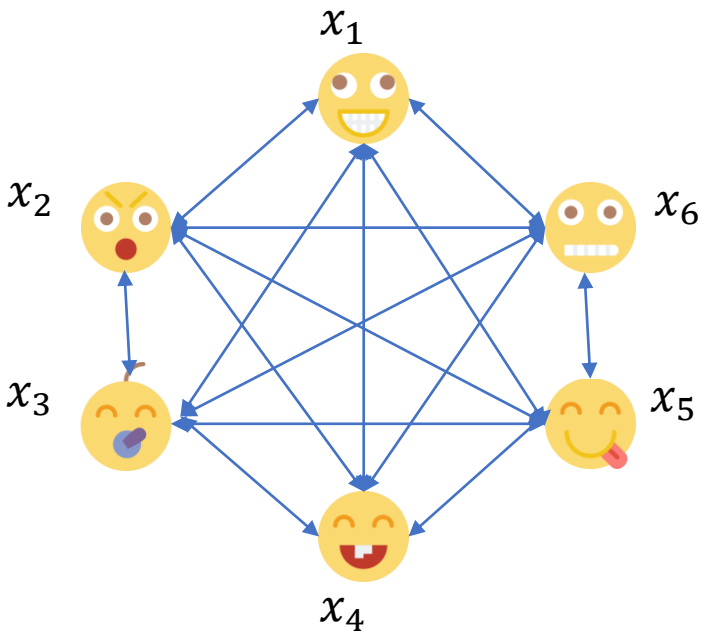
κ

Each party in set has a (signed) share of c_i

MPC

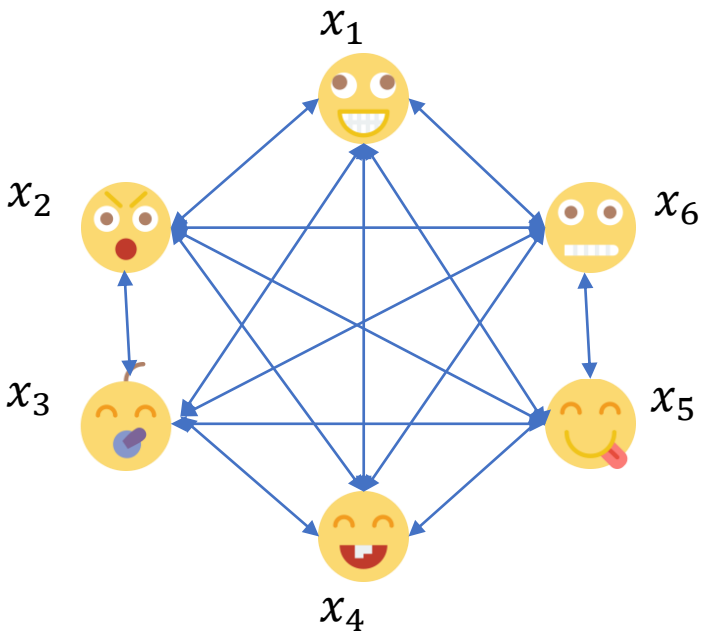
MPC

Multi-Party Computation with ℓ-output quality



MPC

Multi-Party Computation with ℓ-output quality



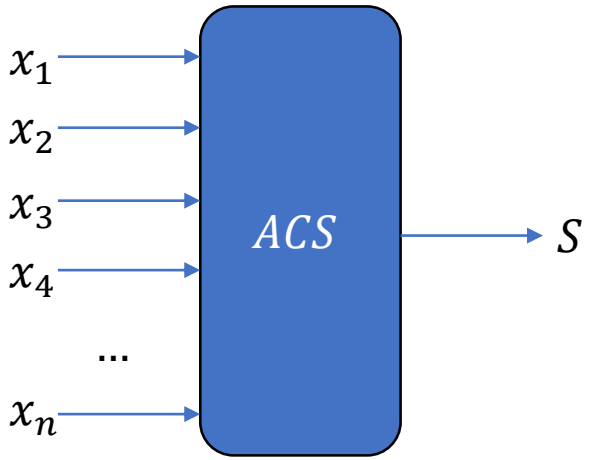
Adversary chooses S with size at least ℓ

$g(x_1', x_2', \ldots, x_n')$, where $x_i' = x_i$ if $P_i \in S$

$x_i' = \perp$ otherwise

MPC

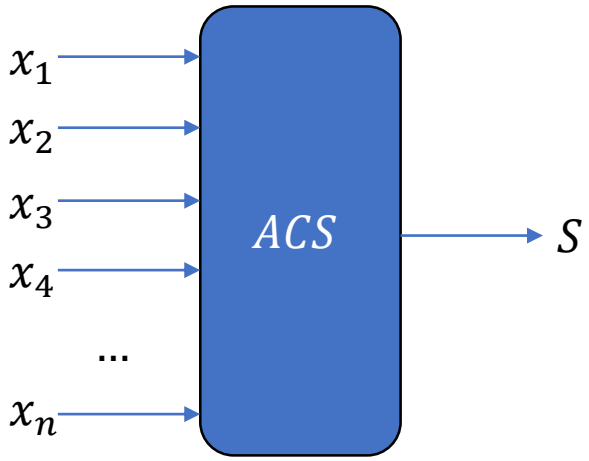
Agreement on a Common Subset with ℓ-output quality



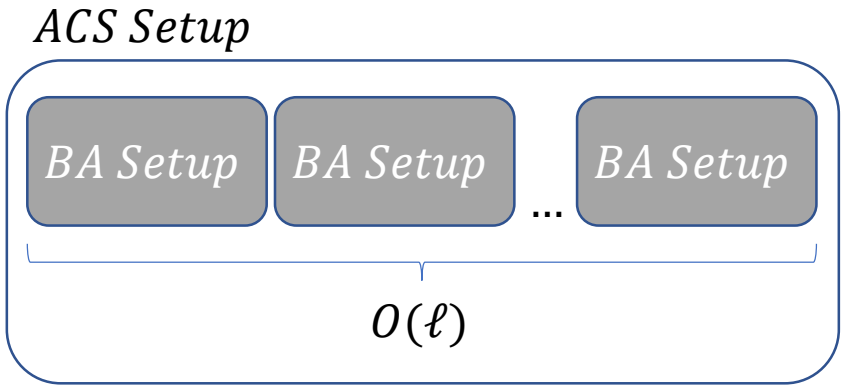
$|S| \geq \ell$ with $\ell - f$ honest inputs

MPC

Agreement on a Common Subset with ℓ-output quality



$|S| \geq \ell$ with $\ell - f$ honest inputs



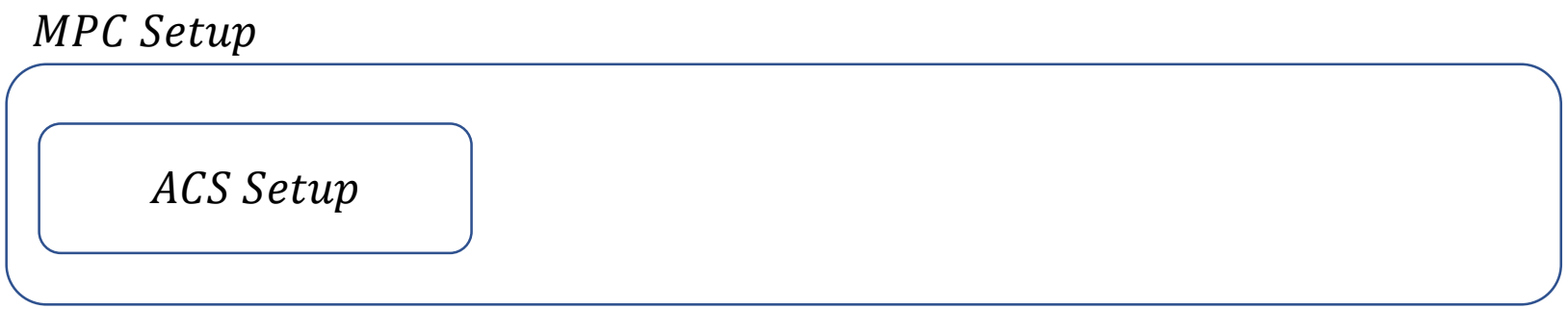
Communication $O(\ell \cdot \mathcal{I} \cdot poly(\kappa) \cdot n)$
Setup size $O\big(\ell \cdot poly(\kappa)\big)$

MPC

Threshold Fully Homomorphic Encryption

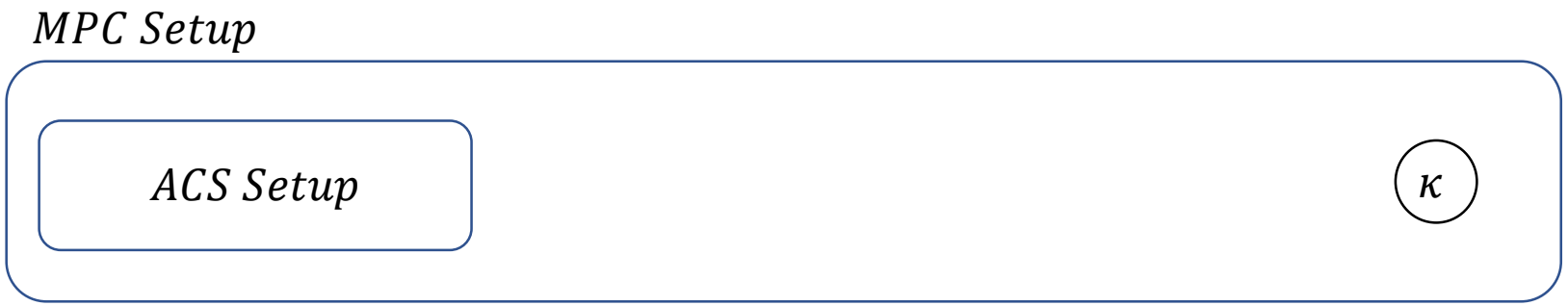
MPC

Threshold Fully Homomorphic Encryption



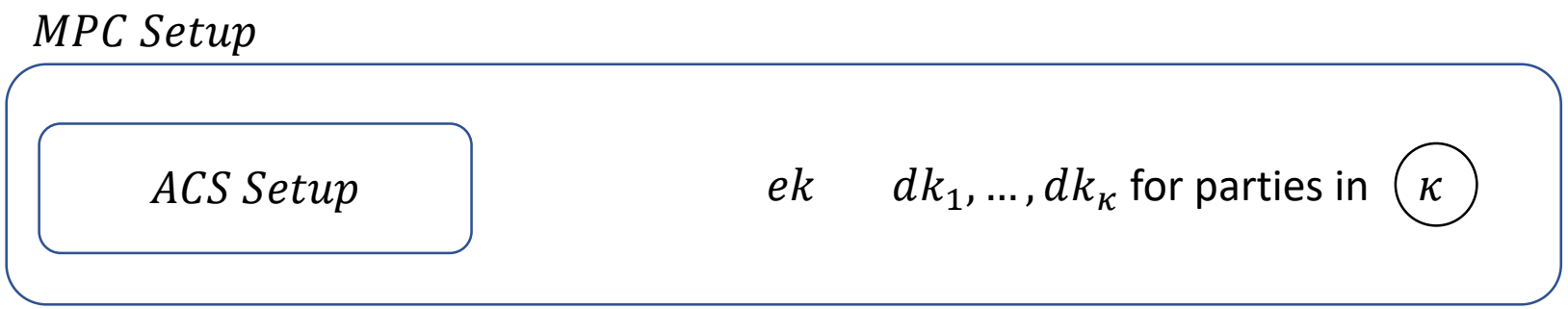
MPC

Threshold Fully Homomorphic Encryption



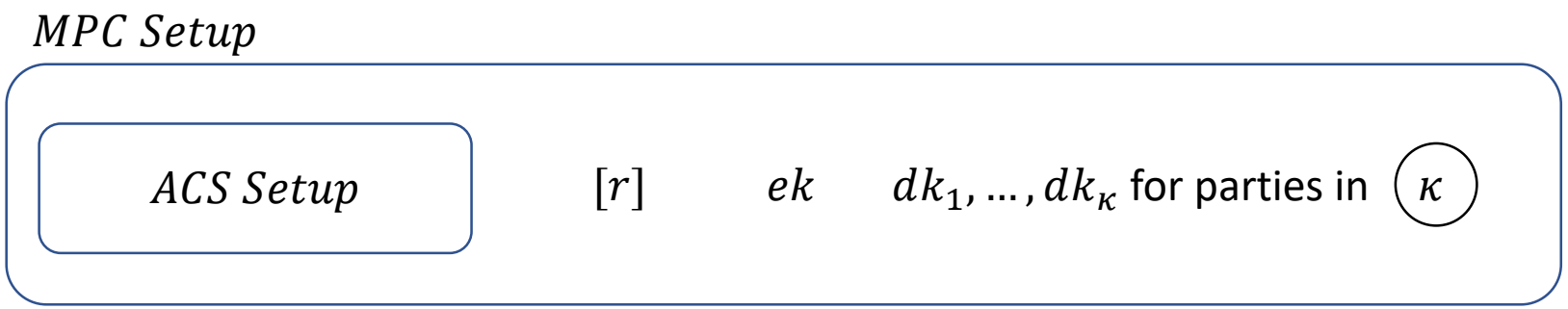
MPC

Threshold Fully Homomorphic Encryption



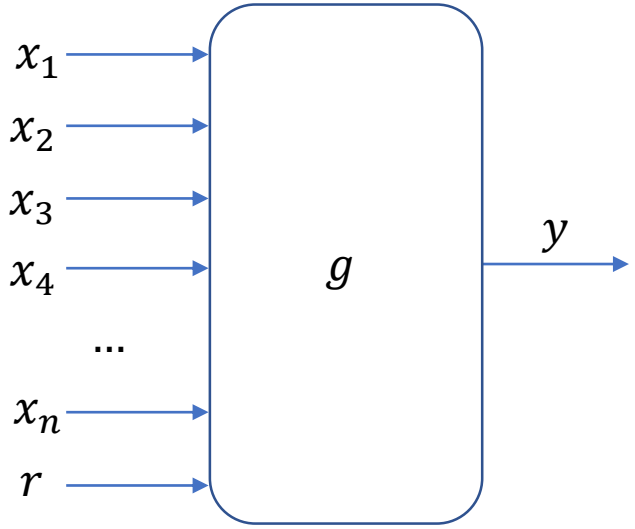
MPC

Threshold Fully Homomorphic Encryption



MPC

Threshold Fully Homomorphic Encryption

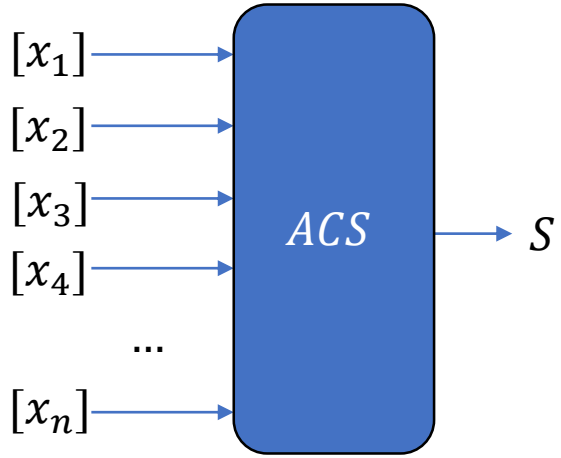


$MPC\ Setup$

$ACS\ Setup$ $[r]$ ek $dk_1, \ldots, dk_\kappa$ for parties in κ

MPC

Threshold Fully Homomorphic Encryption

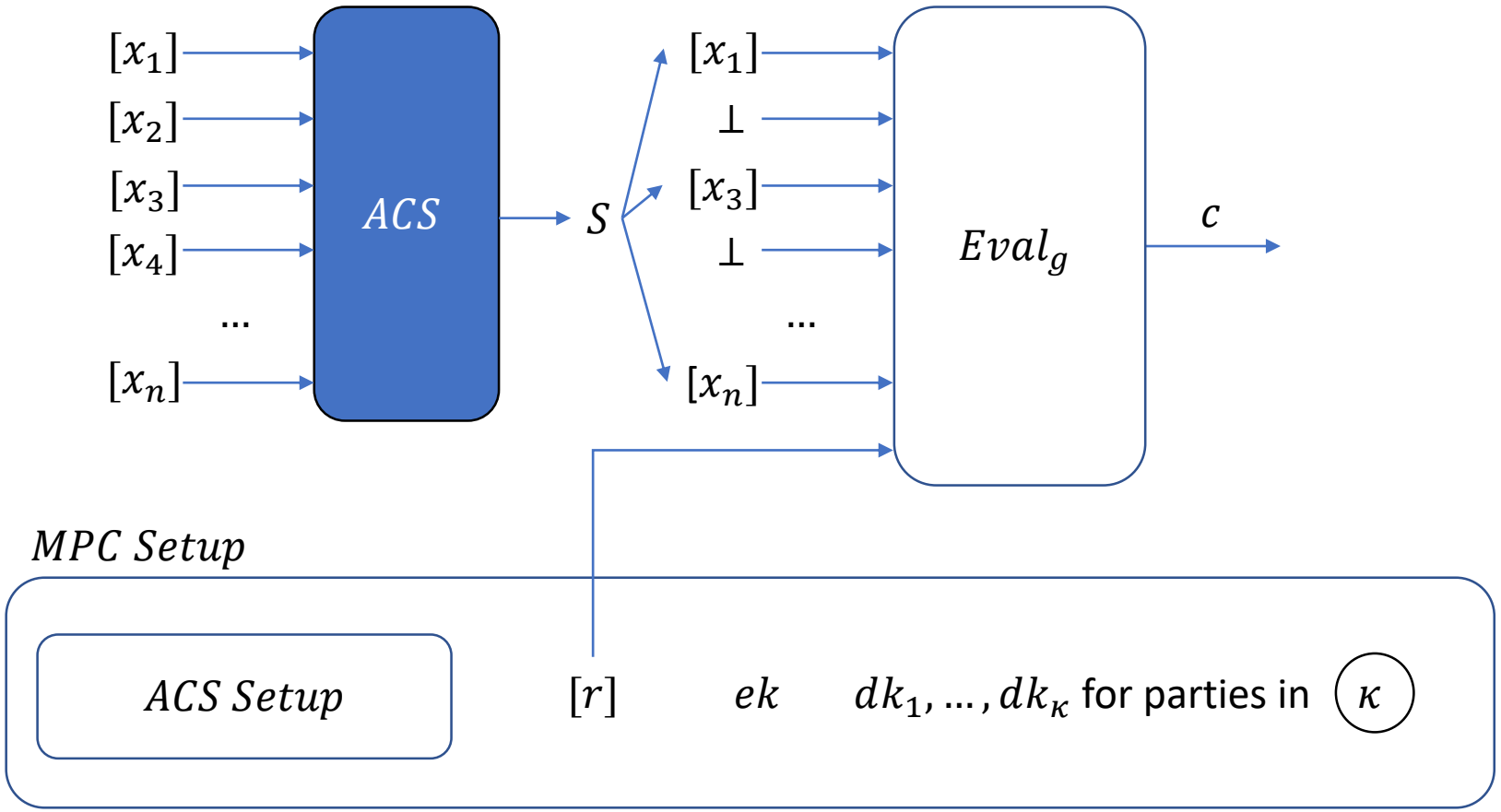


$MPC\ Setup$

$ACS\ Setup$ $[r]$ ek $dk_1, \ldots, dk_\kappa$ for parties in κ

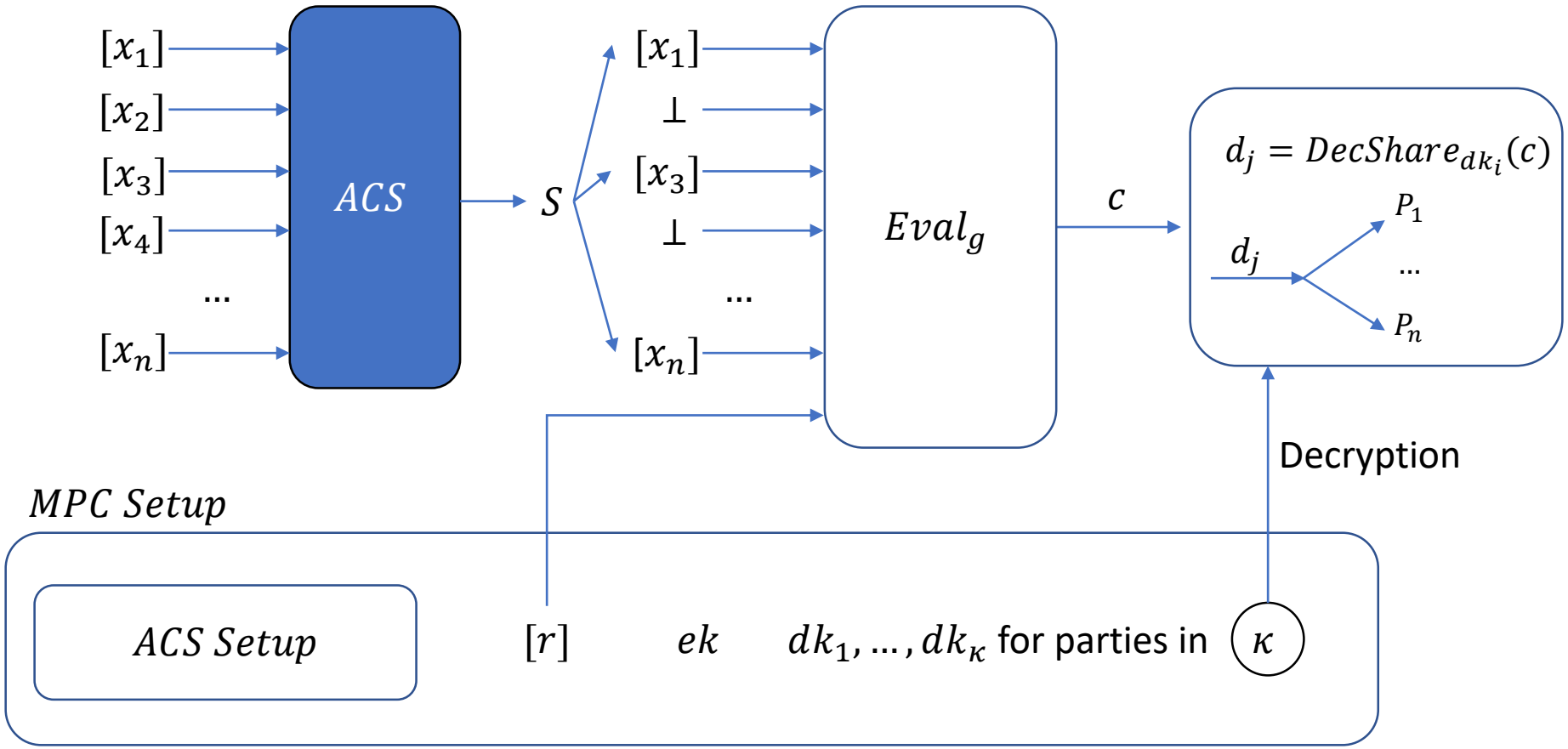
MPC

Threshold Fully Homomorphic Encryption



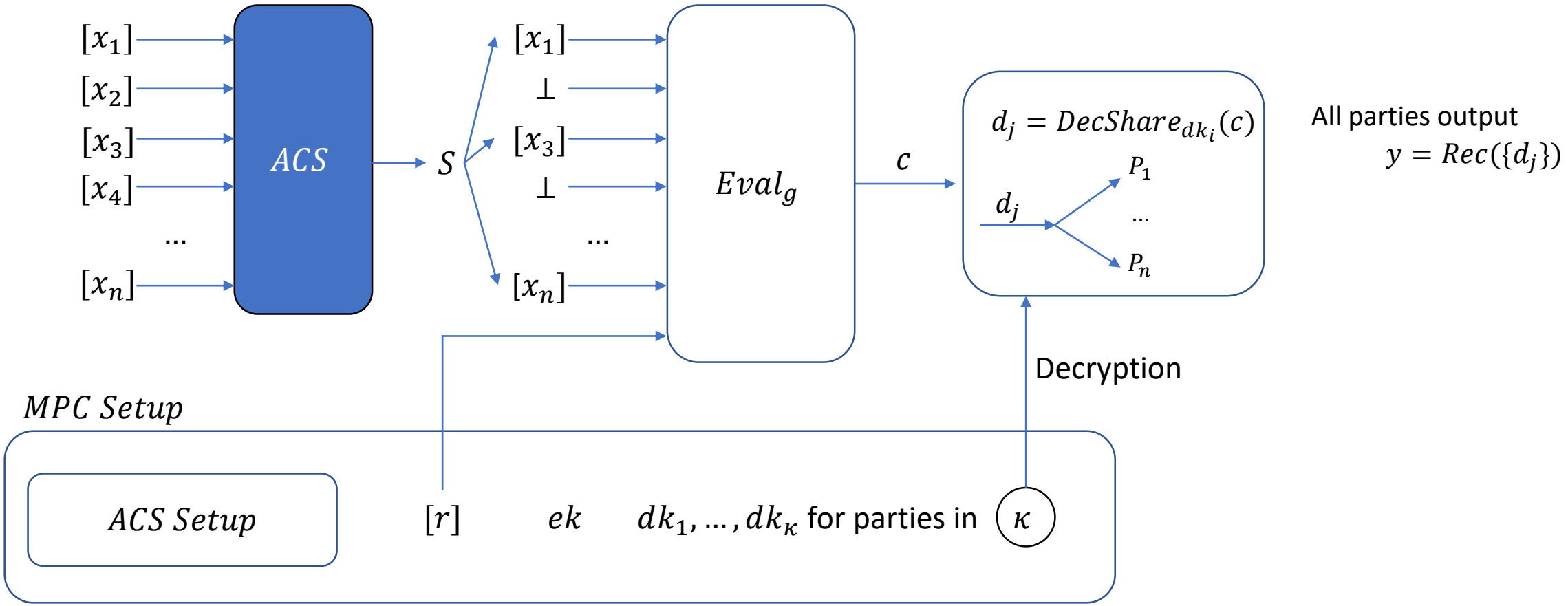
MPC

Threshold Fully Homomorphic Encryption



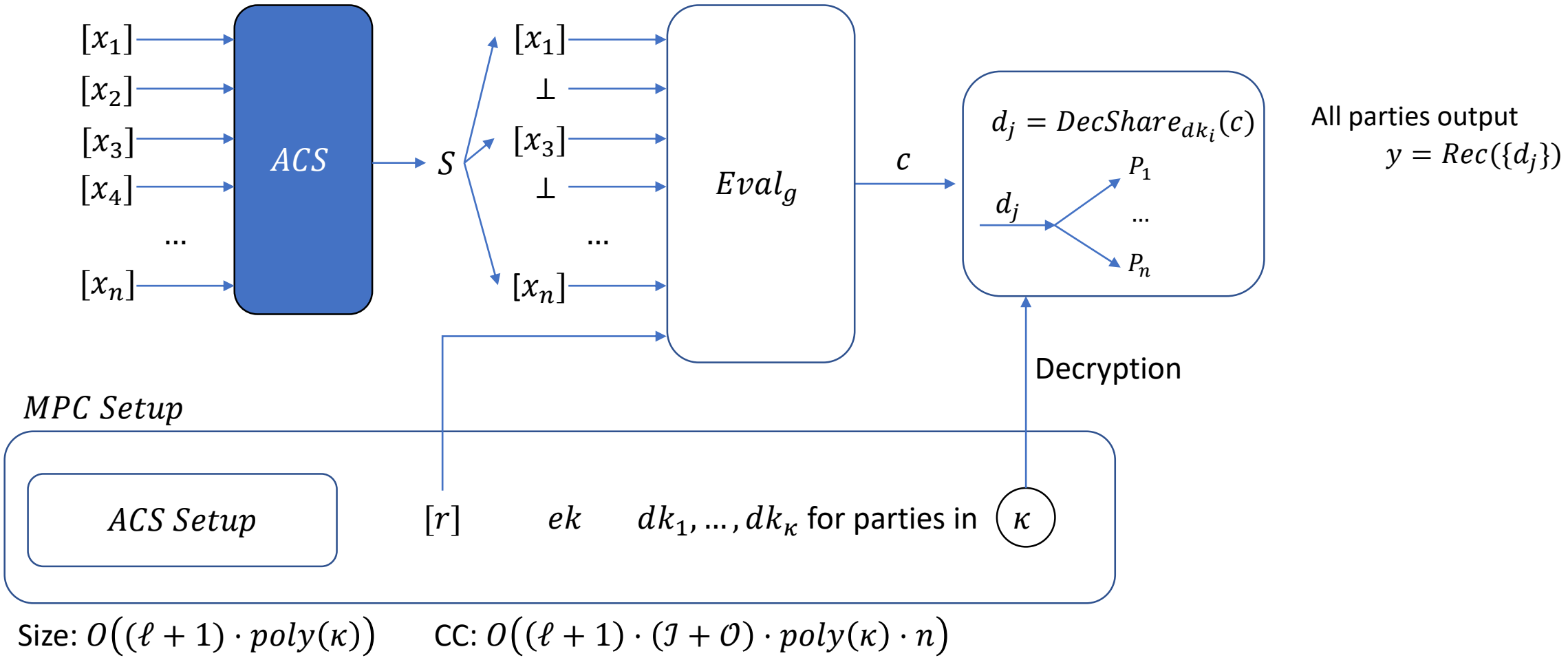
MPC

Threshold Fully Homomorphic Encryption



MPC

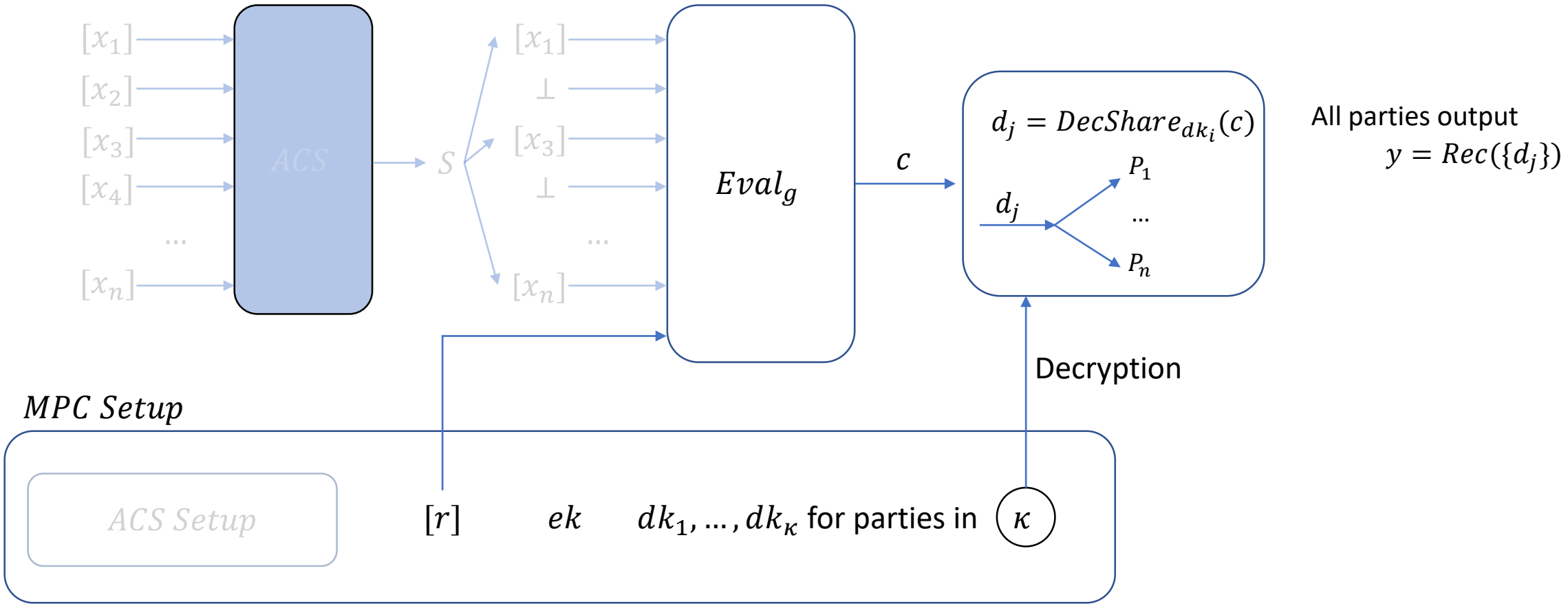
Threshold Fully Homomorphic Encryption



Size: $O\big((\ell + 1) \cdot poly(\kappa)\big)$ CC: $O\big((\ell + 1) \cdot (\mathcal{I} + \mathcal{O}) \cdot poly(\kappa) \cdot n\big)$

MPC for Trusted Dealer

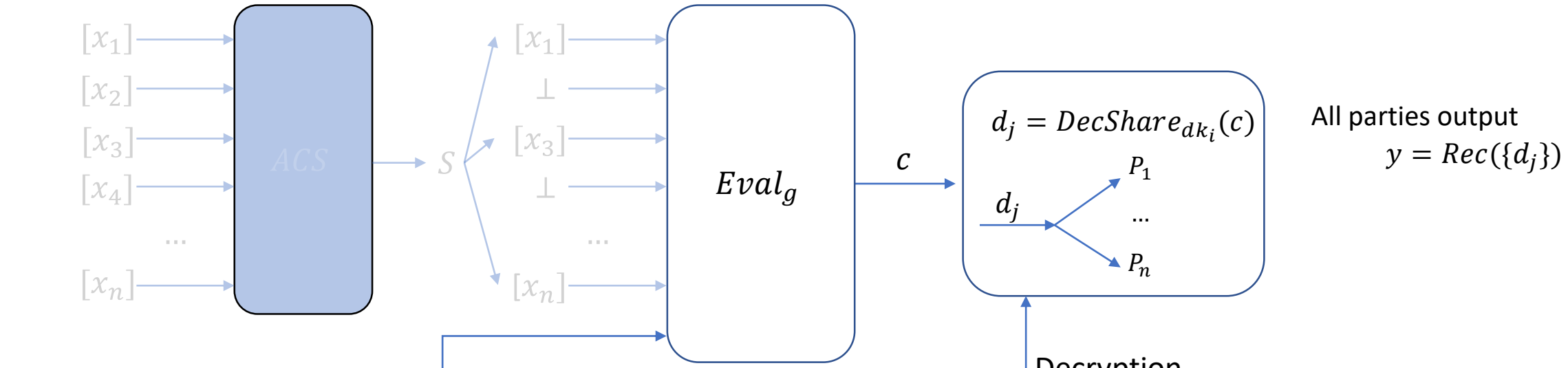
Threshold Fully Homomorphic Encryption



Size: $O\big((\ell + 1) \cdot poly(\kappa)\big)$ CC: $O\big((\ell + 1) \cdot (\mathcal{I} + \mathcal{O}) \cdot poly(\kappa) \cdot n\big)$

MPC for Trusted Dealer

Threshold Fully Homomorphic Encryption

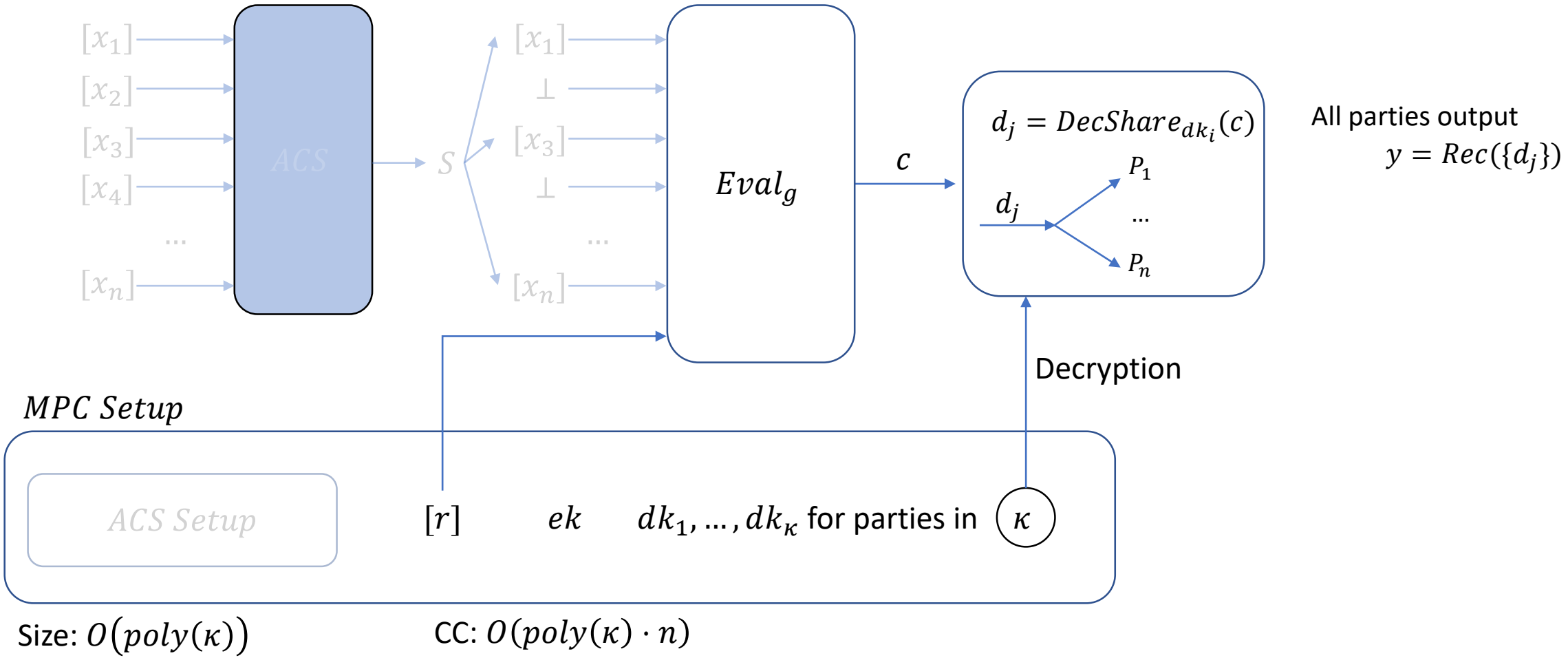


Size: $O\big(poly(\kappa)\big)$

CC: $O\big((\ell + 1) \cdot (\mathcal{I} + \mathcal{O}) \cdot poly(\kappa) \cdot n\big)$

MPC for Trusted Dealer

Threshold Fully Homomorphic Encryption



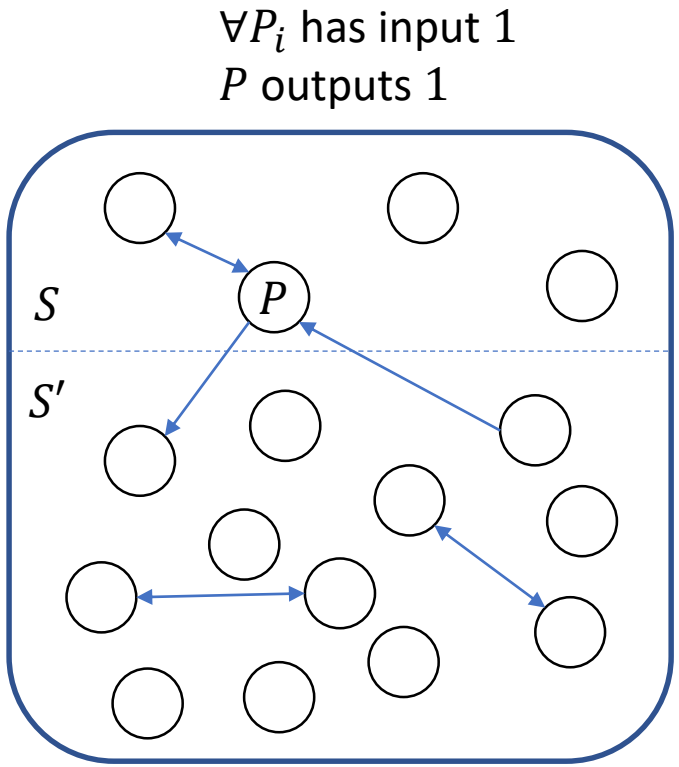
Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

Other lower bounds:

[DR85, A+19] adversary can perform after-the-fact removal

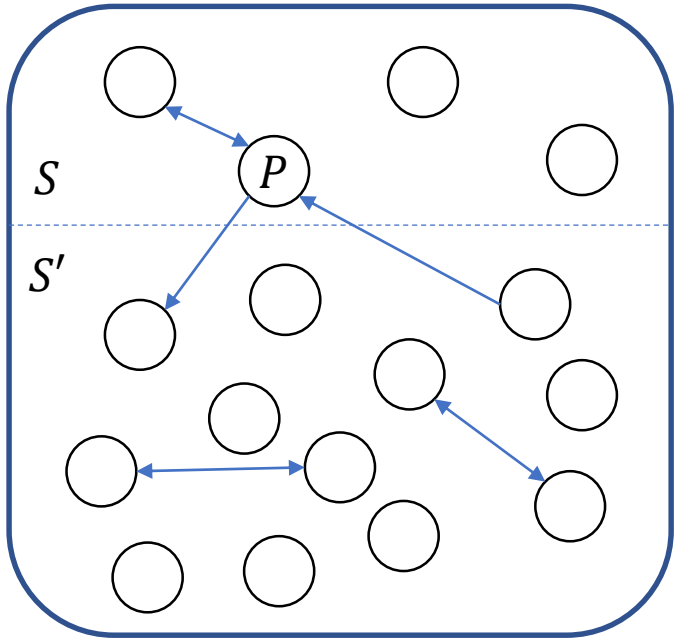
[R20] similar to our lower bound, but with idealized PKI

Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

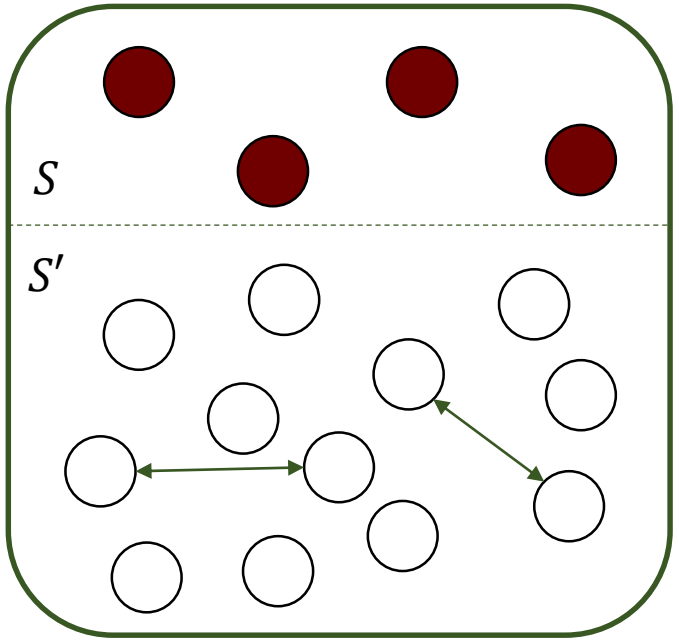


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

$\forall P_i$ has input 1
P outputs 1

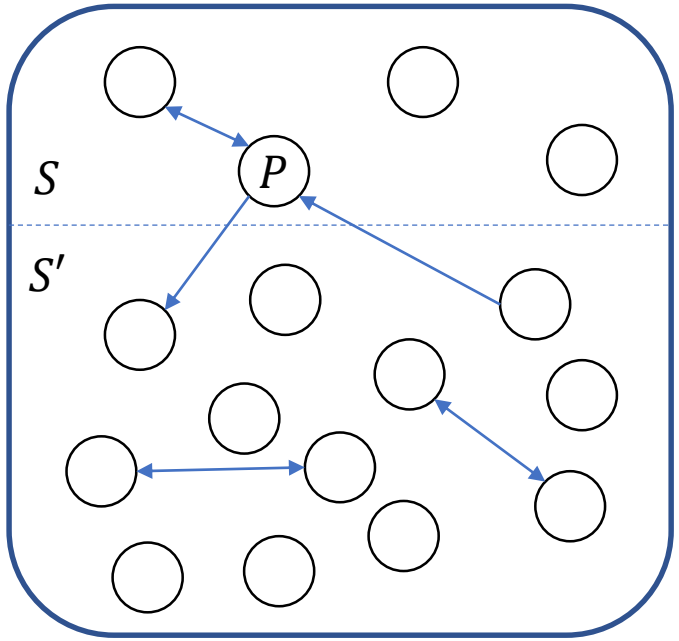


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

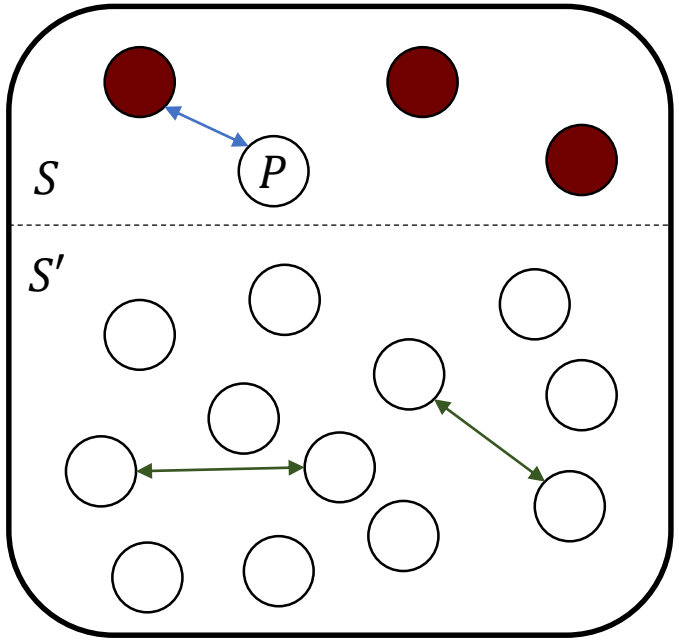


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

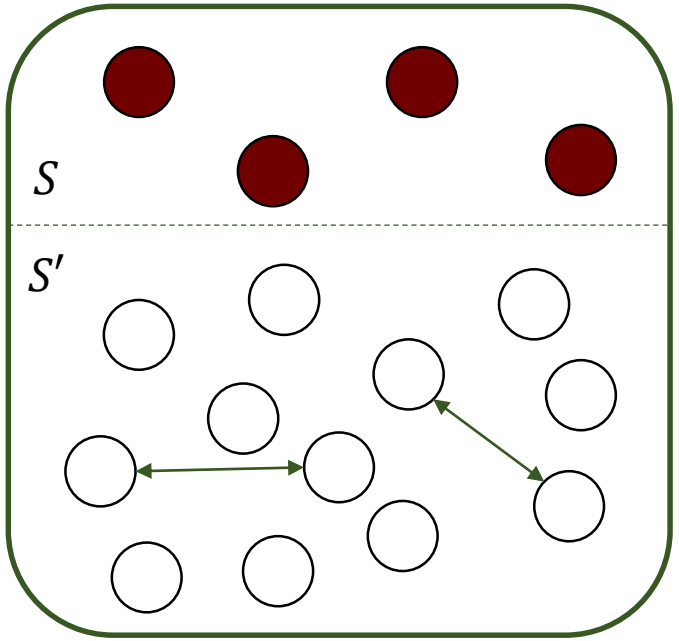
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

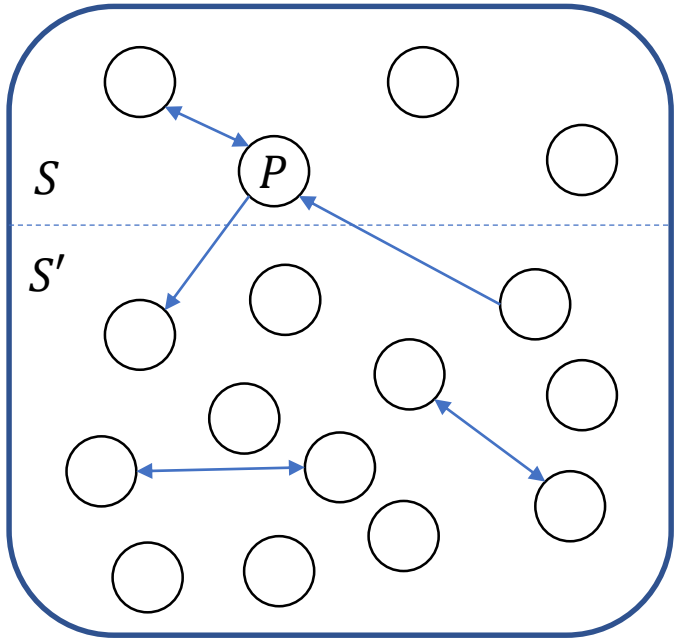


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

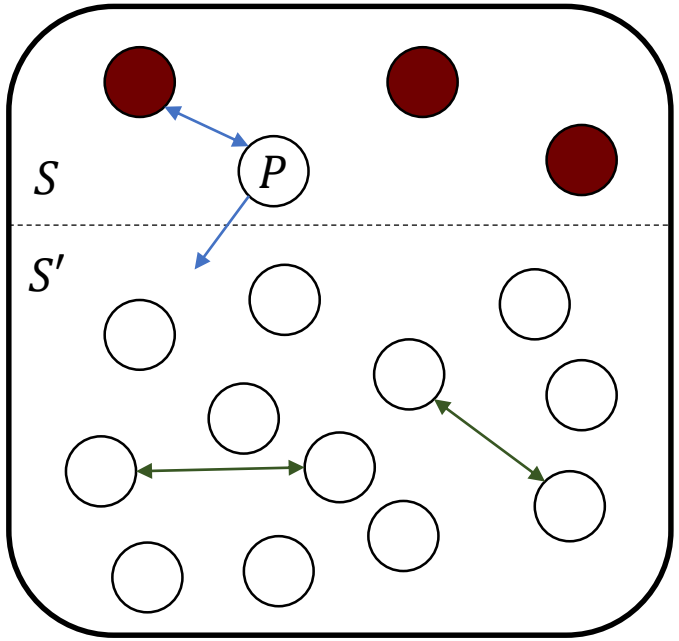


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

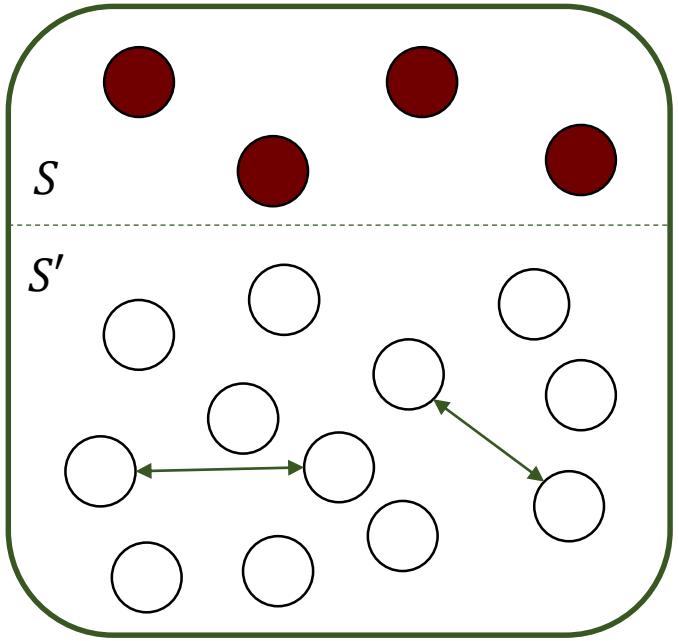
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

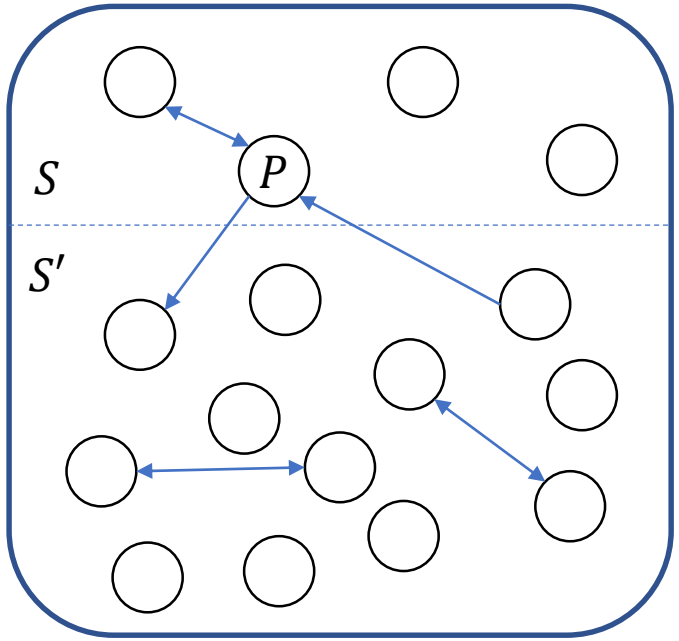


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

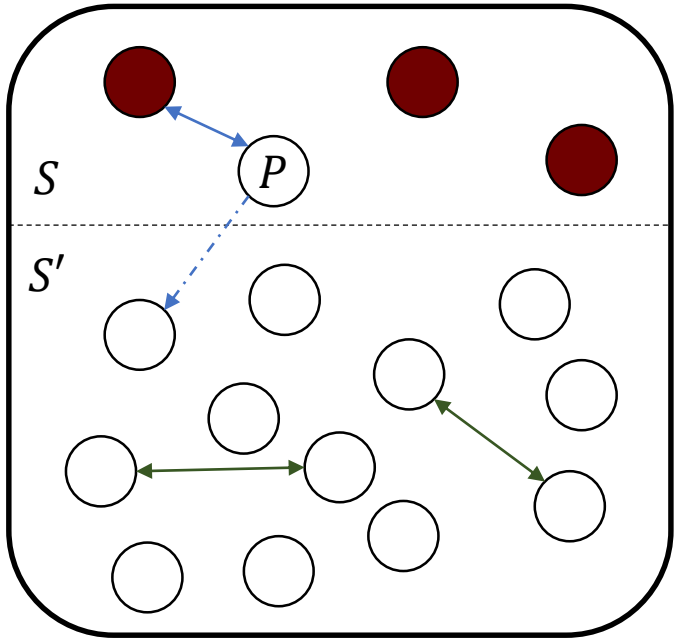


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

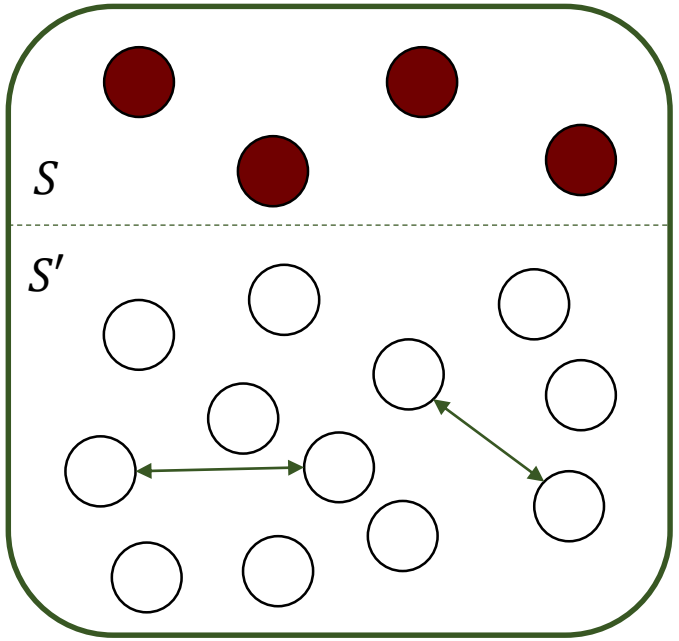
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

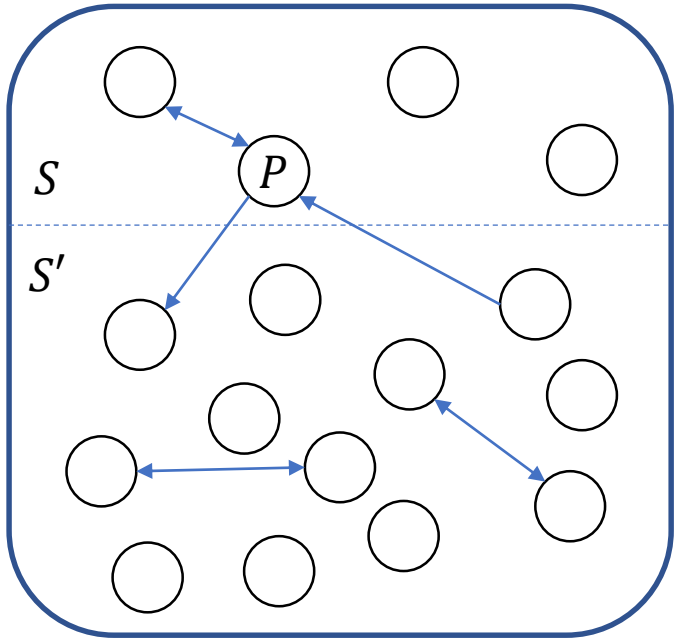


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

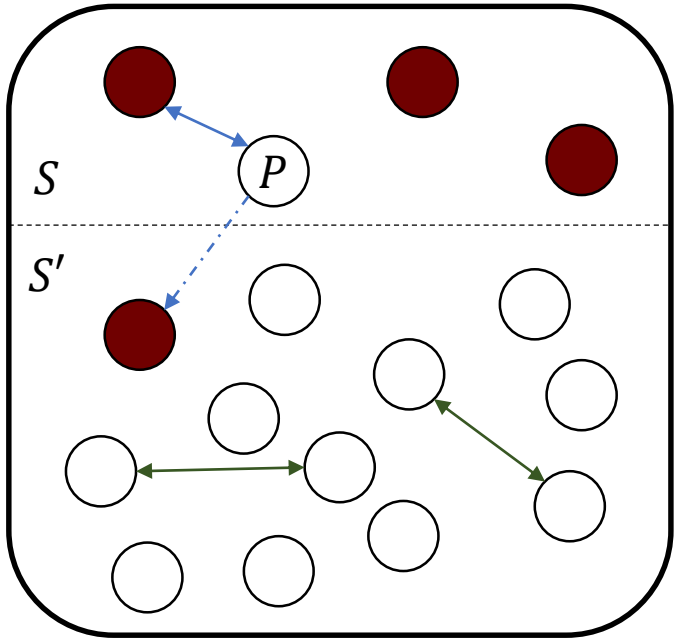


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

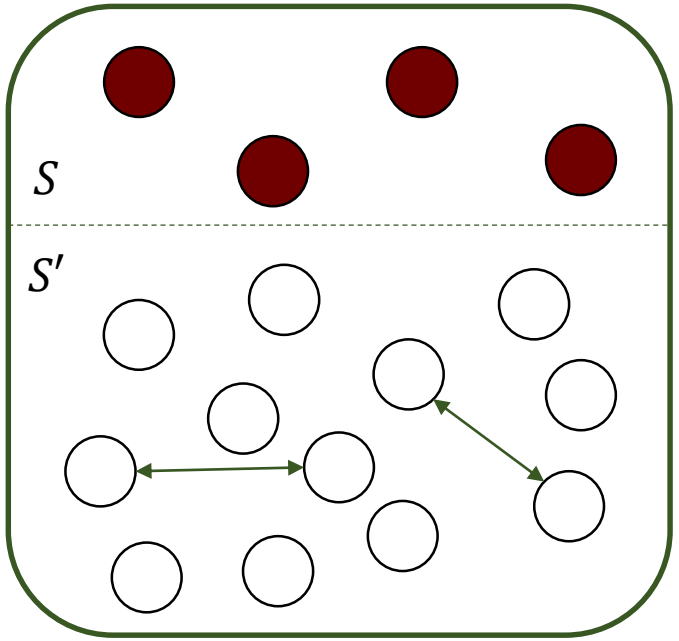
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

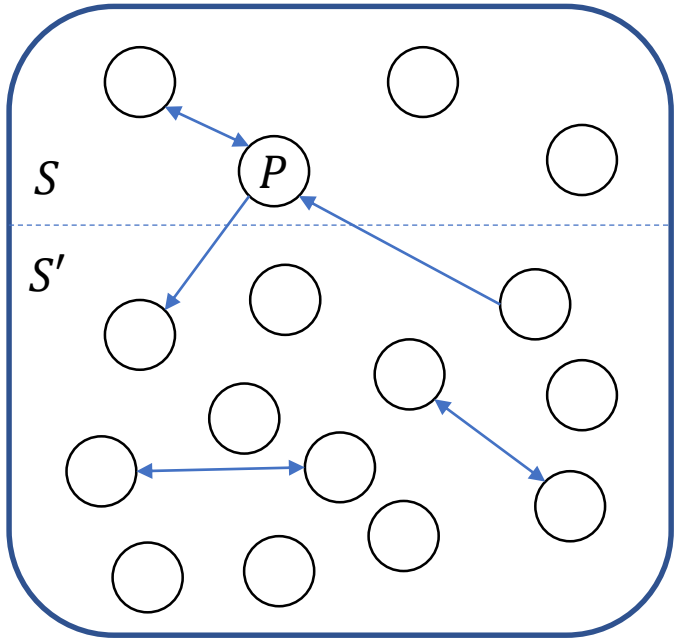


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

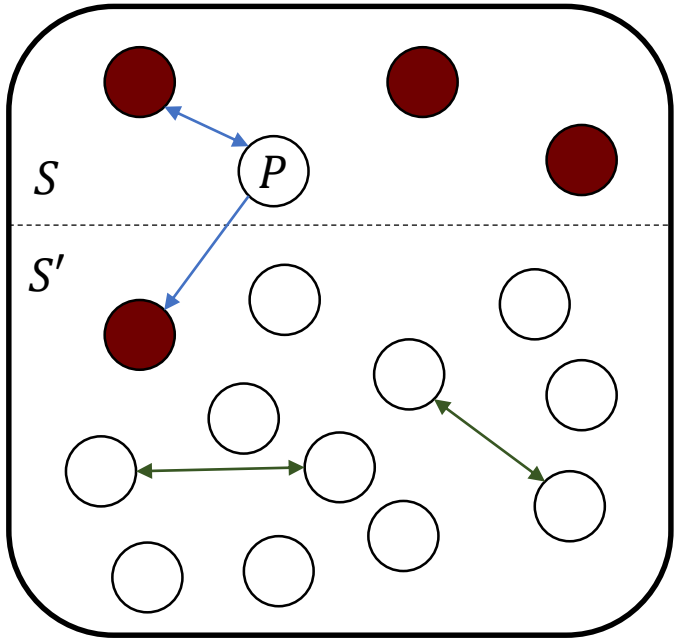


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

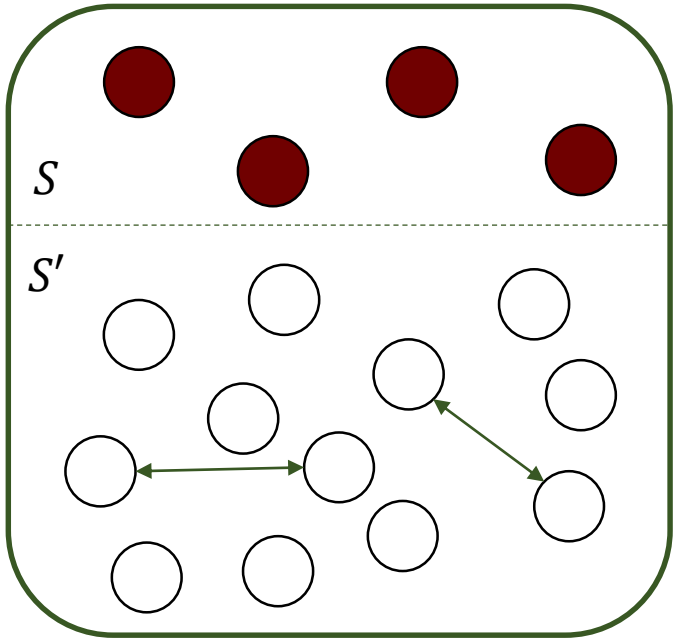
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

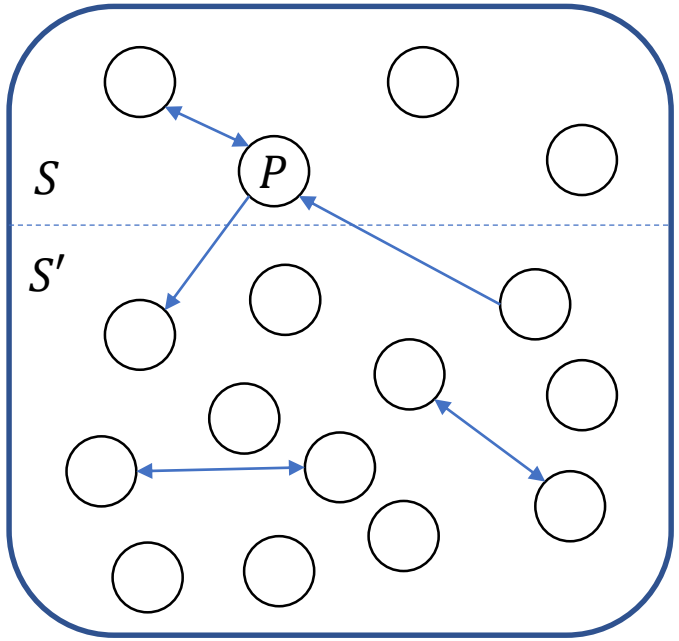


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

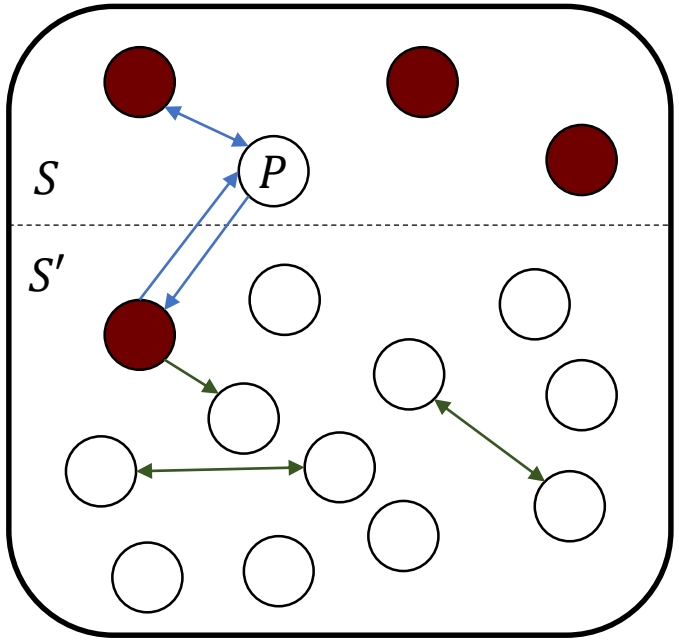


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

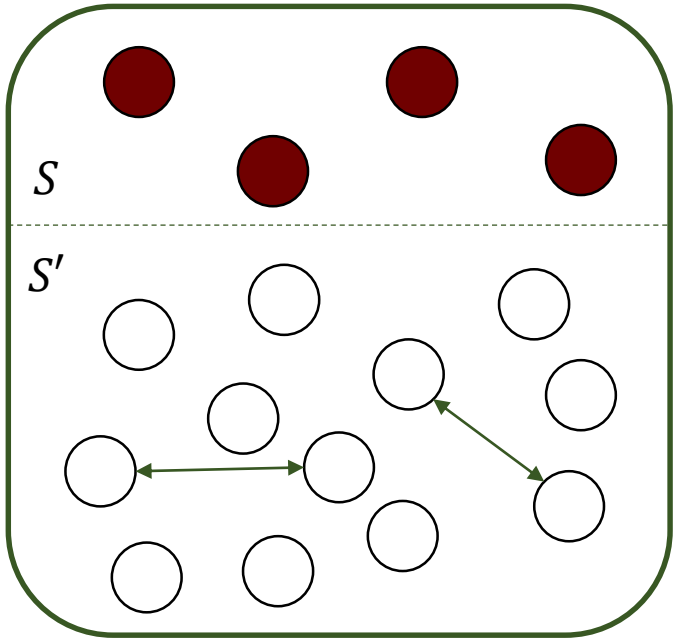
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

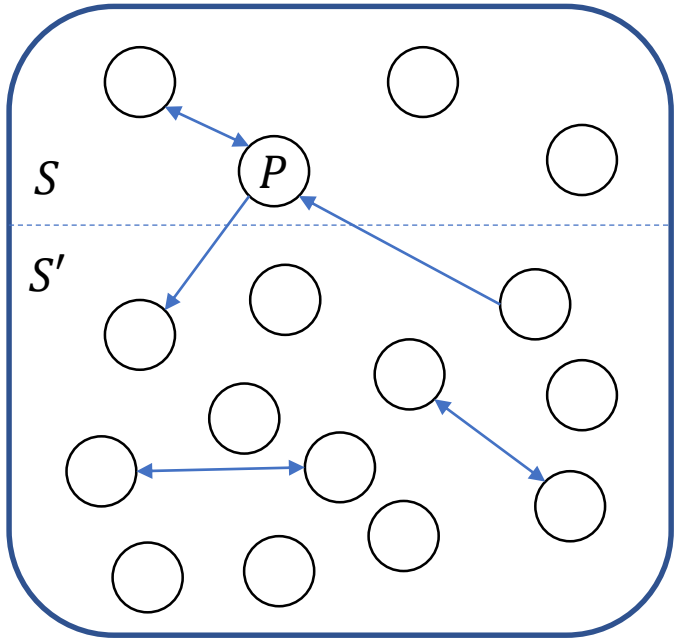


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

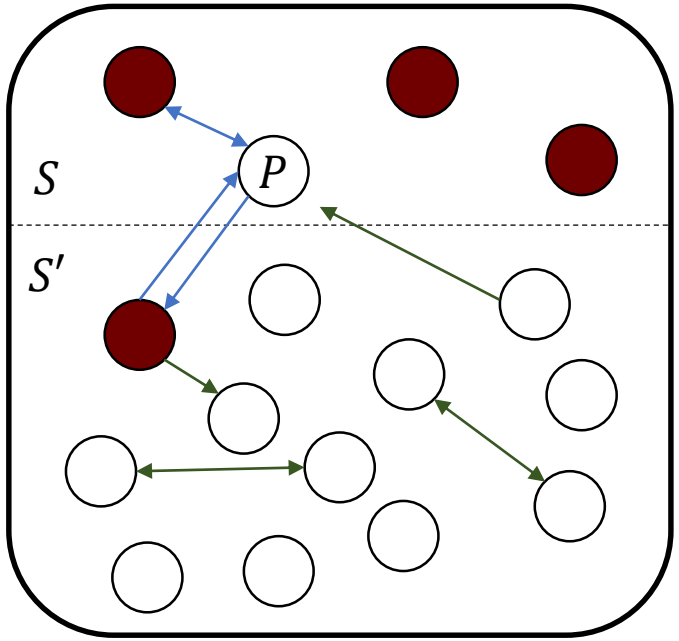


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

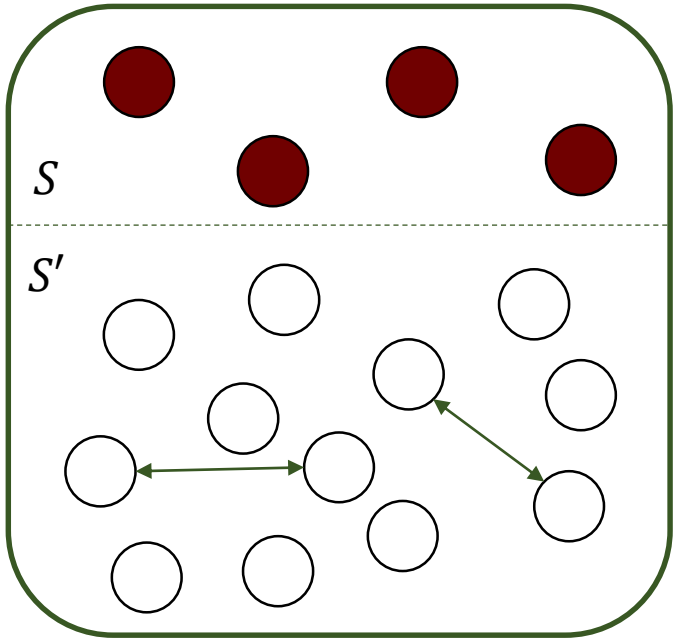
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

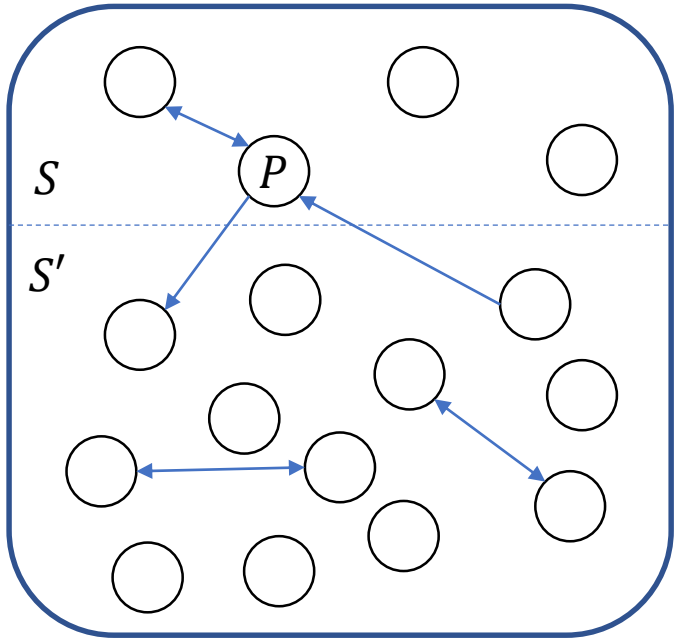


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

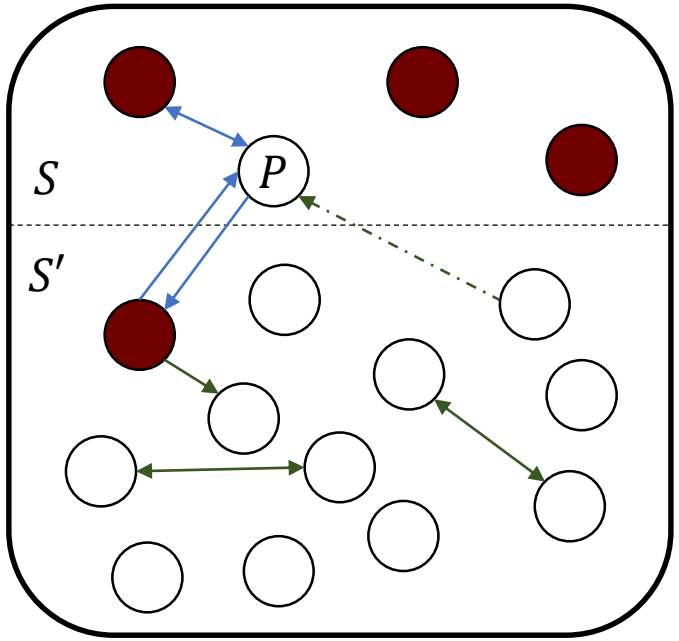


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

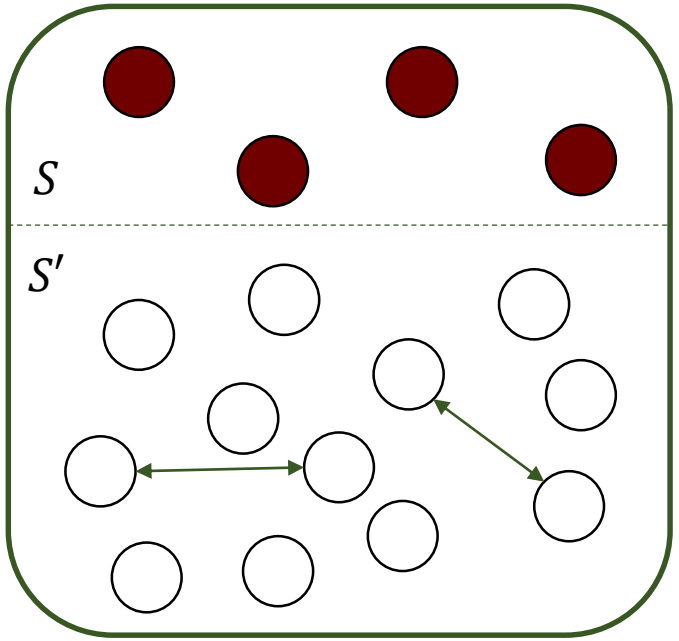
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

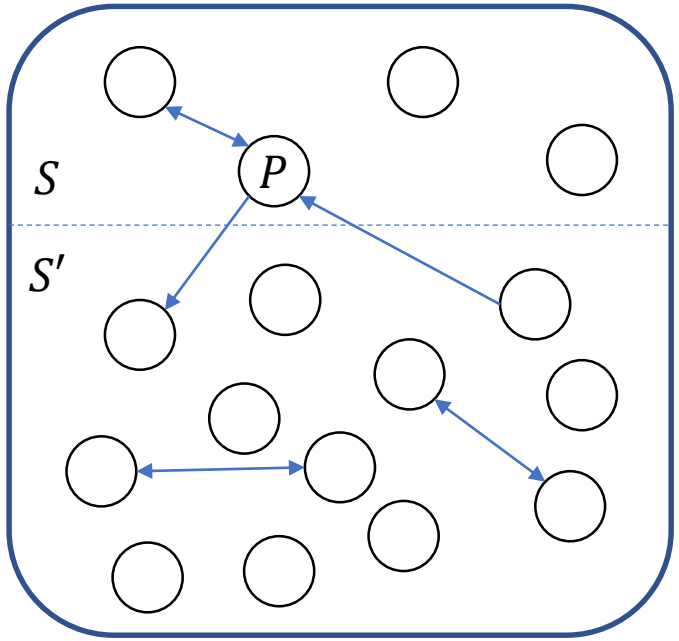


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

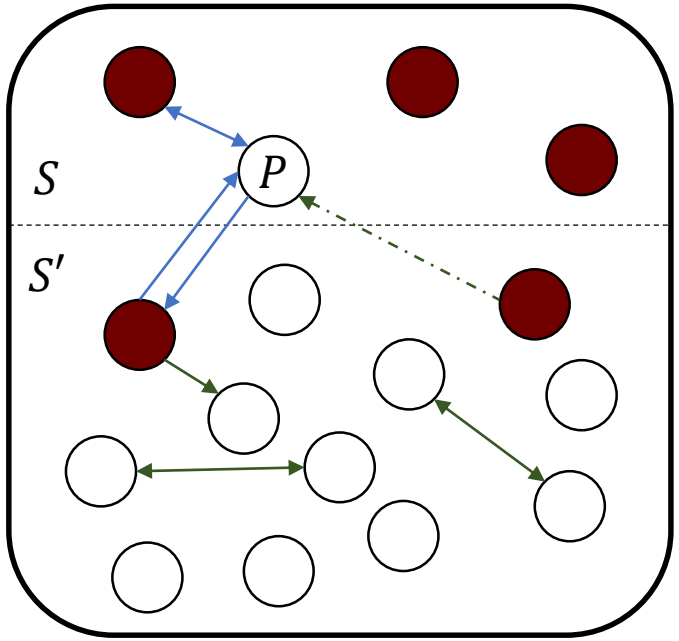


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

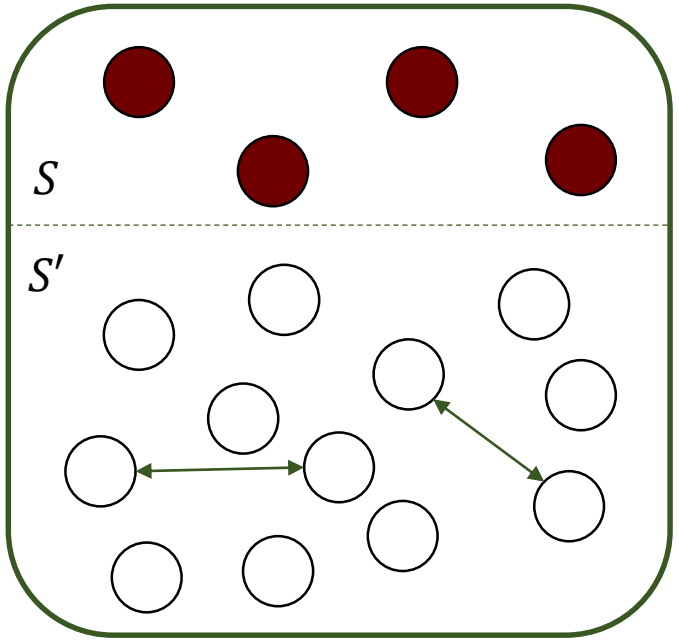
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

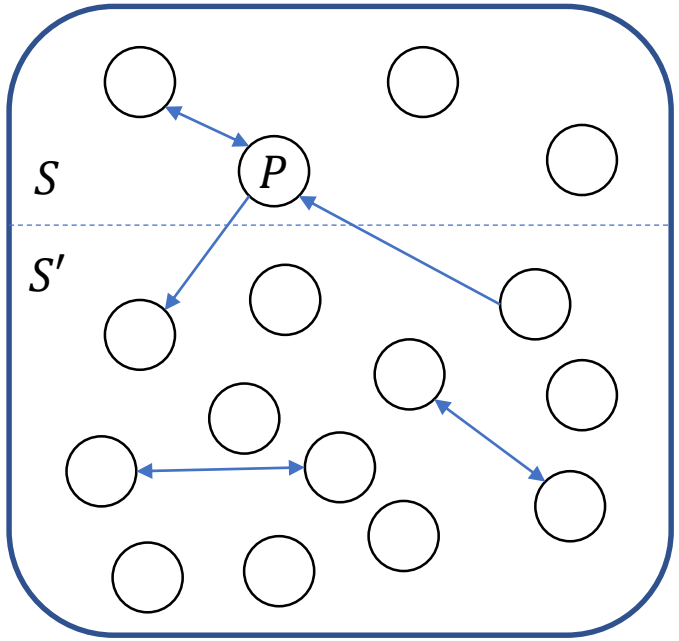


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

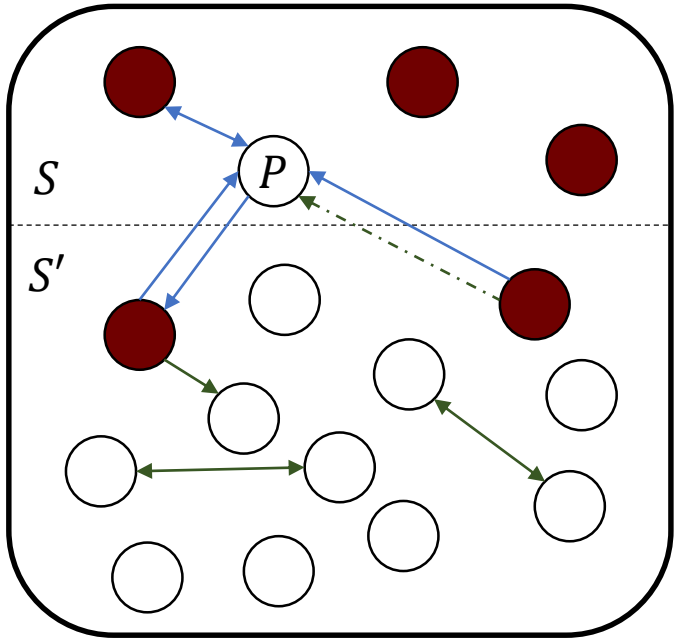


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

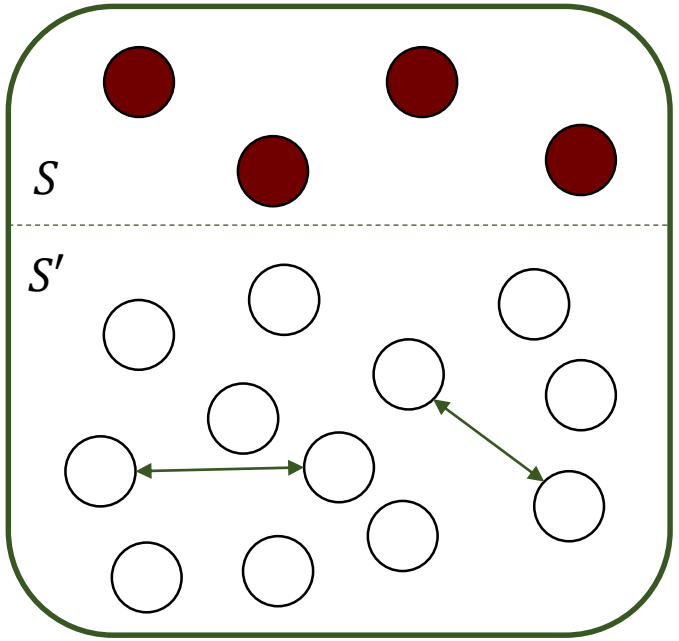
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0

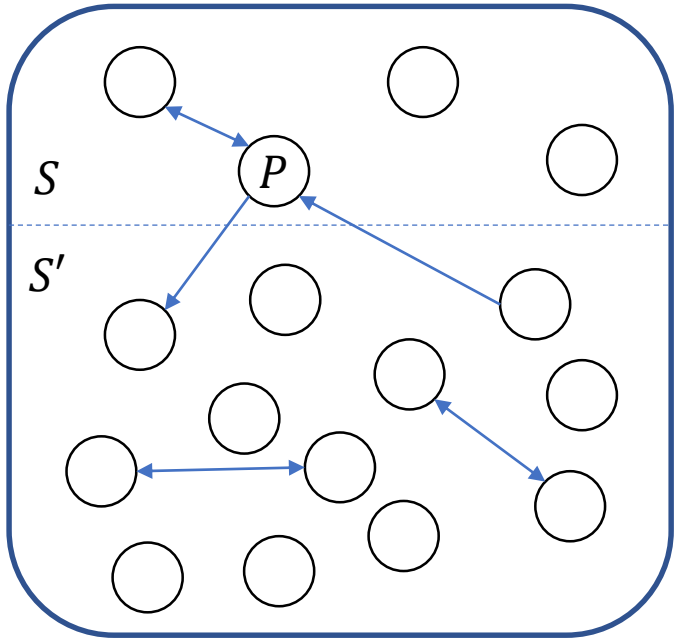


$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0

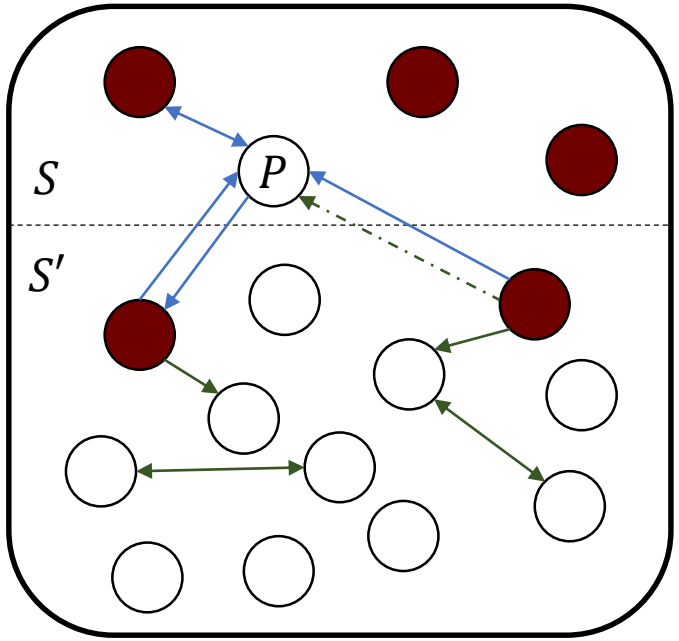


Impossibility of asynch. $o(n^2)$ BA with $\theta(n)$ adaptive corruptions and no setup

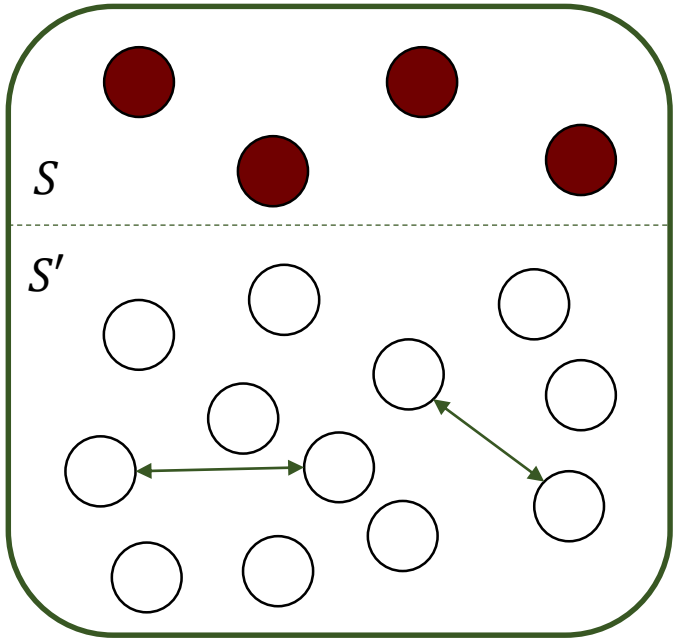
$\forall P_i$ has input 1
P outputs 1



P has input 1; $\forall P_i \in S'$ has input 0
P outputs 1; $\forall P_i \in S'$ outputs 0



$\forall P_i \in S'$ has input 0
$\forall P_i \in S'$ outputs 0



References and Credits

Full version: https://eprint.iacr.org/2020/851

References:

[BKLL20]: Ran Canetti and Tal Rabin. Fast asynchronous Byzantine agreement with optimal resilience. STOC 1993.

[DR85]: Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for Byzantine agreement. Journal of the ACM 1985.

[KS06]: Valerie King, Jared Saia, Vishal Sanwalani, and Erik Vee. Scalable leader election. SODA 2006.

[KS10]: Valerie King and Jared Saia. Breaking the $O(n^2)$ bit barrier: scalable byzantine agreement with an adaptive adversary. PODC 2010.

[M17]: Silvio Micali. Very simple and efficient byzantine agreement. ITCS 2017.

[A+19]: Ittai Abraham, T.-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. PODC 2019.

[CKS20]: Shir Cohen, Idit Keidar, and Alexander Spiegelman. Not a COINcidence: Sub-quadratic asynchronous Byzantine agreement WHP. DISC 2020.

[R20]: Matthieu Rambaud. Lower bounds for authenticated randomized Byzantine consensus under (partial) synchrony: The limits of standalone digital signatures.

Credits:

Icons: https://www.flaticon.com/