Efficient Adaptively-Secure Byzantine Agreement for Long Messages

Kartik Nayak

with Amey Bhangale, Chen-Da Liu-Zhang, Julian Loss

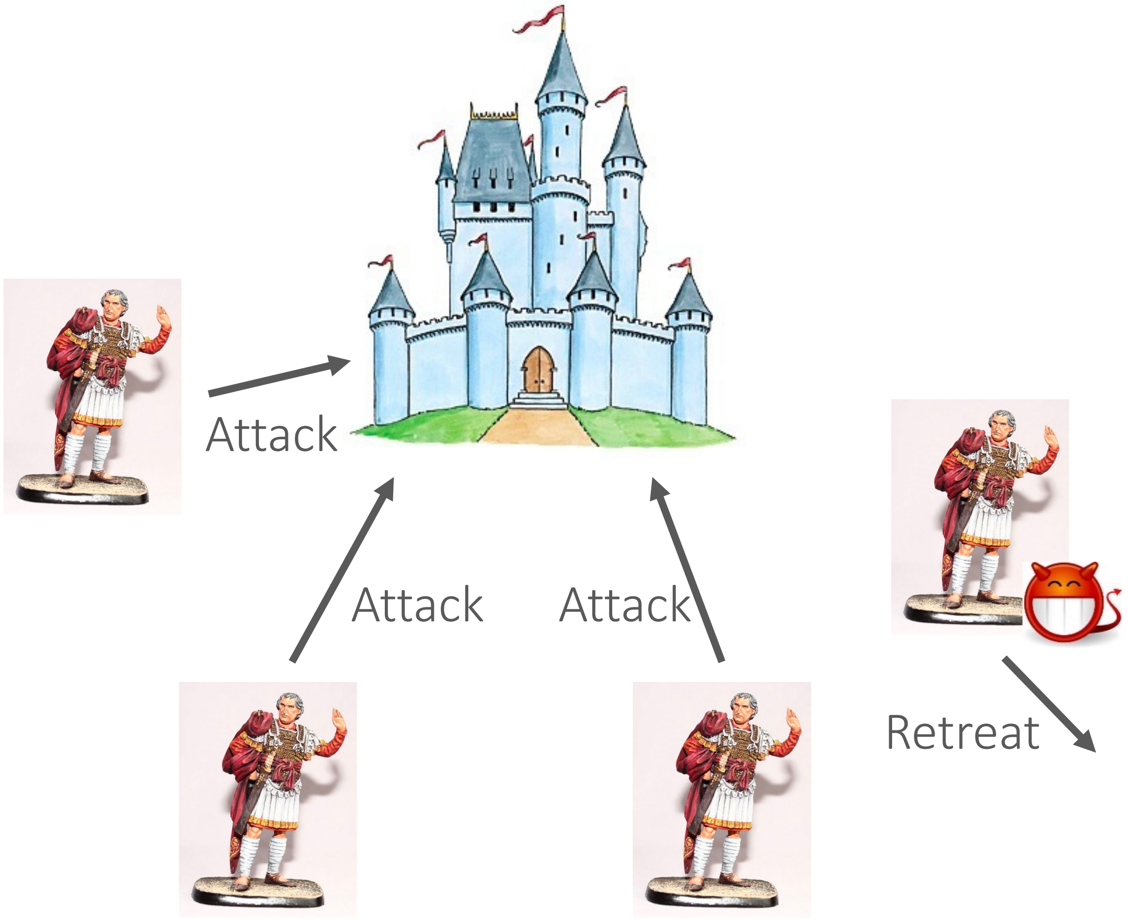
Duke UNIVERSITY

Byzantine Agreement

n generals (≤ t Byzantine) need to agree on a battle plan

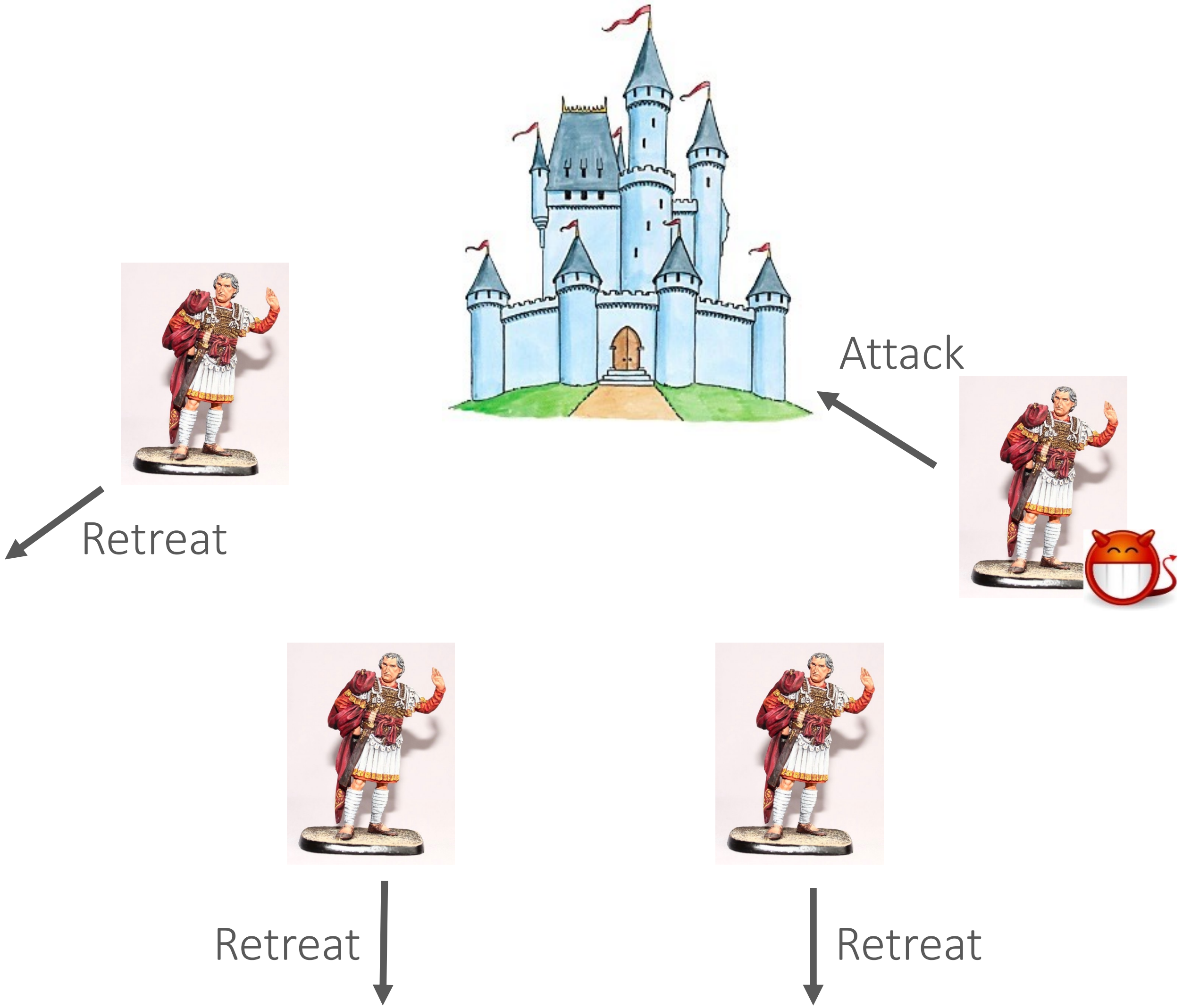
Byzantine Agreement

n generals ($\leq$ t Byzantine) need to agree on a battle plan



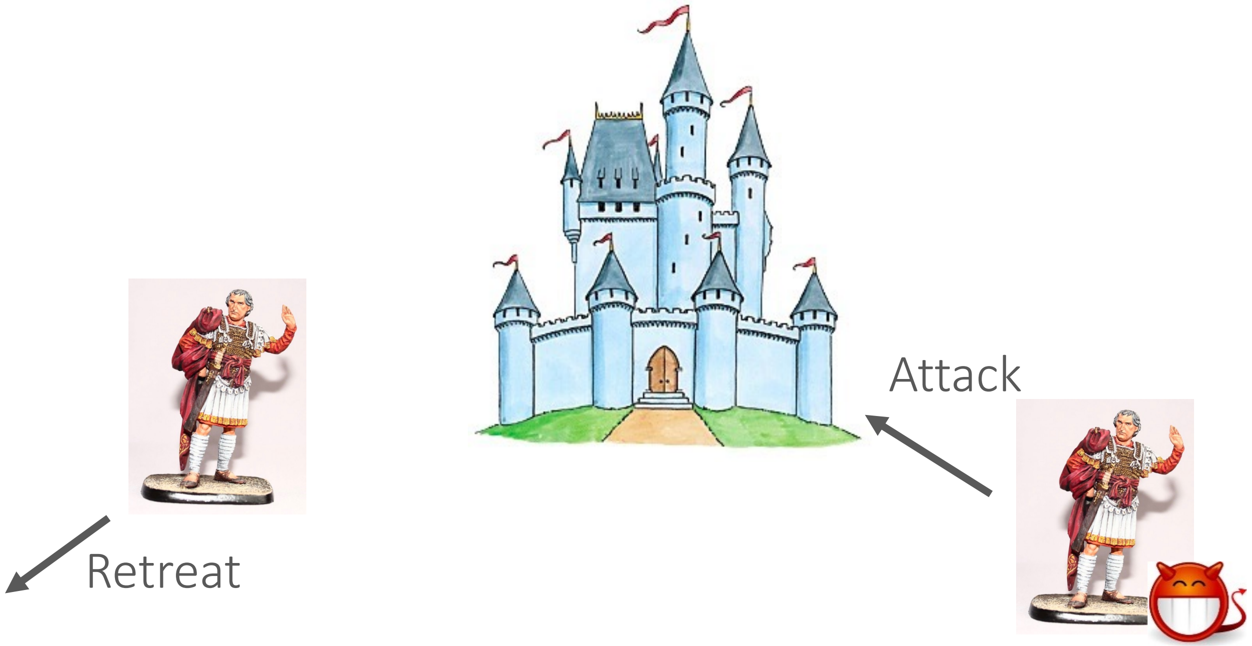
Byzantine Agreement

n generals (≤ t Byzantine) need to agree on a battle plan



Byzantine Agreement

n generals (≤ t Byzantine) need to agree on a battle plan



Requirements:
- Agreement: no two honest generals disagree
- Validity: if all generals start with same input, they commit that input
- Termination

Some Key Properties For BA Protocols

Some Key Properties For BA Protocols

1. Communication complexity
2. Security under adaptive adversaries

Goal: Can we achieve a BA protocol with "low communication complexity" while being secure under an adaptive adversary?

Bound on Communication Complexity [DR'82]

Bound on Communication Complexity [DR'82]

Dolev-Reischuk bound: Any deterministic BA protocol needs honest parties to send $\Omega(t^2)$ messages

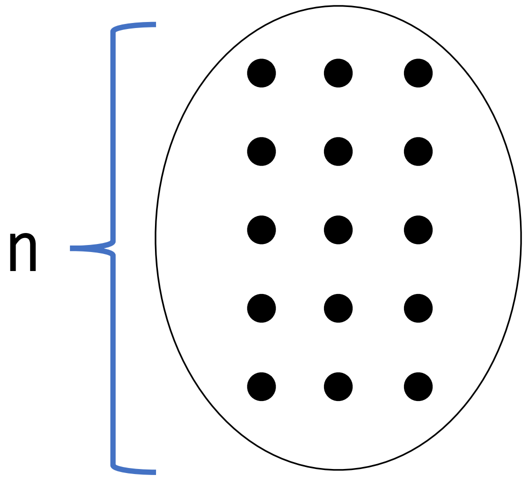
- Typically $t = O(n)$, so $\Omega(n^2)$ messages

Can we achieve BA with $o(n^2)$ messages?

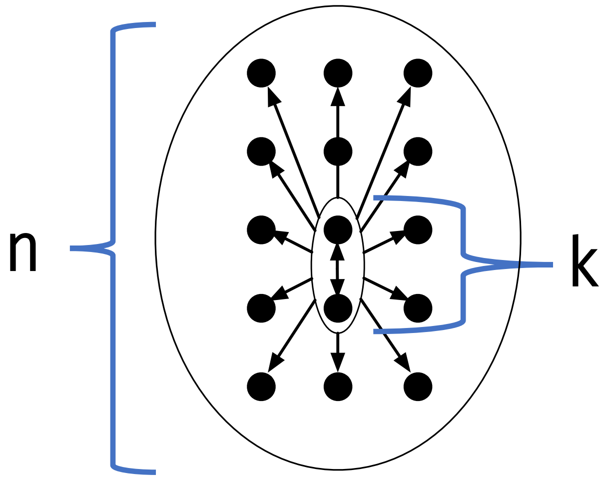
Yes, use randomization!

A Protocol with Sub-Quadratic Messages

A Protocol with Sub-Quadratic Messages

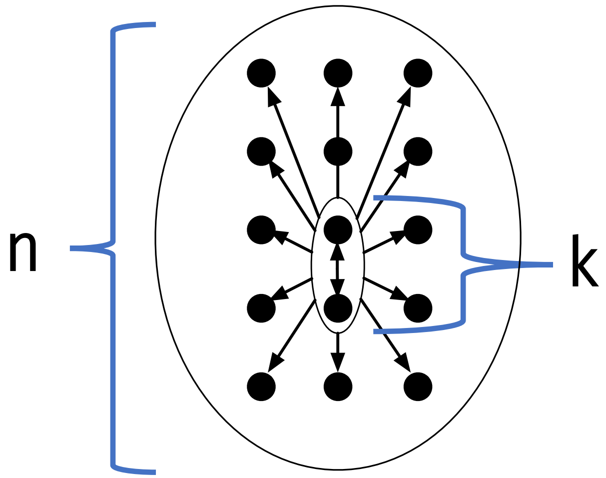


A Protocol with Sub-Quadratic Messages



A Protocol with Sub-Quadratic Messages

Idea: randomly elect a small committee of size k



Only the committee members send messages to all parties; thus, communication = O(poly(k).n)

Tolerating Adaptive Adversary using Player-Replaceability [CM'16]

Tolerating Adaptive Adversary using Player-Replaceability [CM'16]

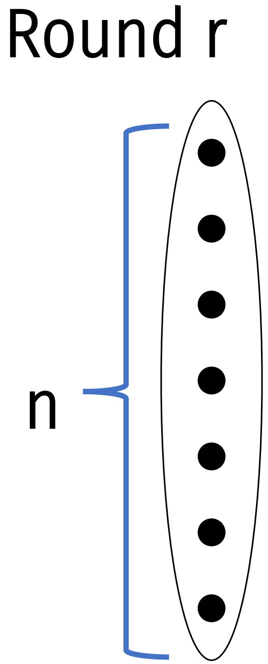
Concern: an adaptive adversary can corrupt the committee

Solution: Player-replaceability, i.e., keep changing the committee after every round

Tolerating Adaptive Adversary using Player-Replaceability [CM'16]

Concern: an adaptive adversary can corrupt the committee

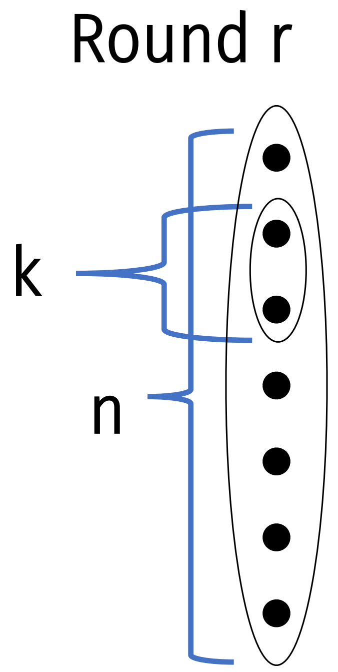
Solution: Player-replaceability, i.e., keep changing the committee after every round



Tolerating Adaptive Adversary using Player-Replaceability [CM'16]

Concern: an adaptive adversary can corrupt the committee

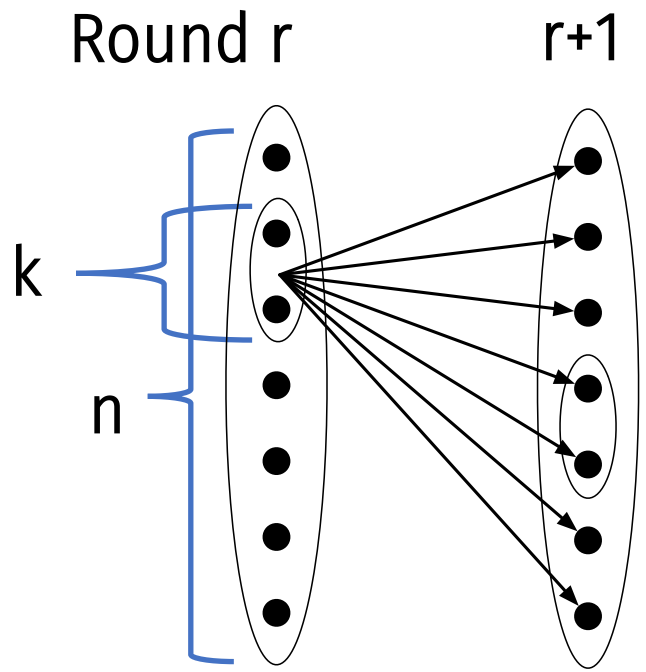
Solution: Player-replaceability, i.e., keep changing the committee after every round



Tolerating Adaptive Adversary using Player-Replaceability [CM'16]

Concern: an adaptive adversary can corrupt the committee

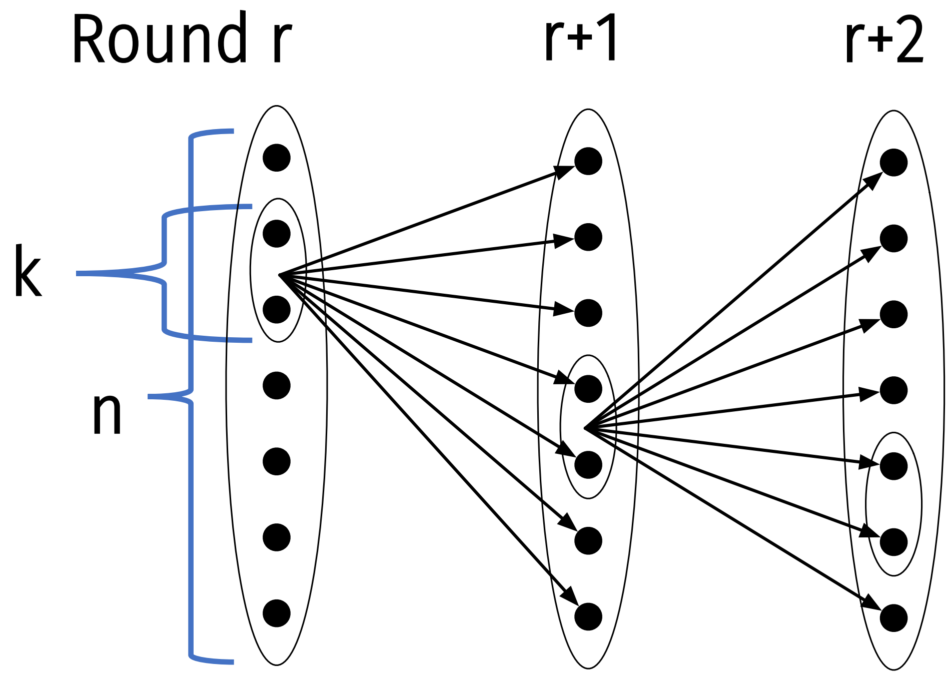
Solution: Player-replaceability, i.e., keep changing the committee after every round



Tolerating Adaptive Adversary using Player-Replaceability [CM'16]

Concern: an adaptive adversary can corrupt the committee

Solution: Player-replaceability, i.e., keep changing the committee after every round



Communication Complexity of BA

Communication Complexity of BA

Thus, we have a BA protocol with O(poly(k).n) messages. Are we done?

If we have an l-bit value, communication complexity is O(poly(k).nl) bits

What happens if l is large?

- e.g., $l = \Omega(n^2)$
- e.g., l = 10 MB sized block

BA Extension Protocols for l-bit Inputs [NRSVX'20]

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires $O(kn^2)$ communication
- Share the l-bit input using erasure coding techniques

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

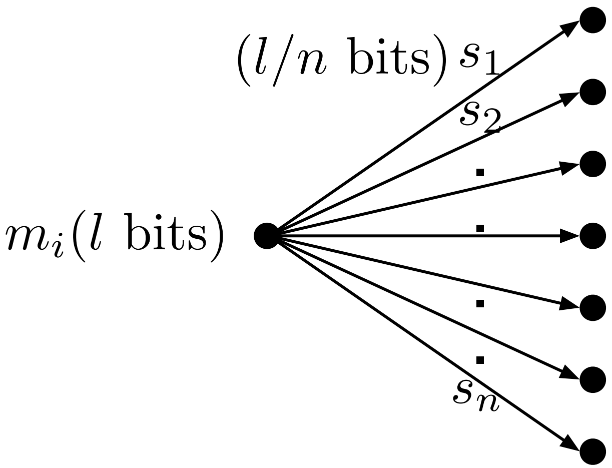
- Agree on a k-bit accumulator value corresponding to one of the inputs, requires $O(kn^2)$ communication
- Share the l-bit input using erasure coding techniques

$m_i(l \text{ bits})$ •

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires $O(kn^2)$ communication
- Share the l-bit input using erasure coding techniques

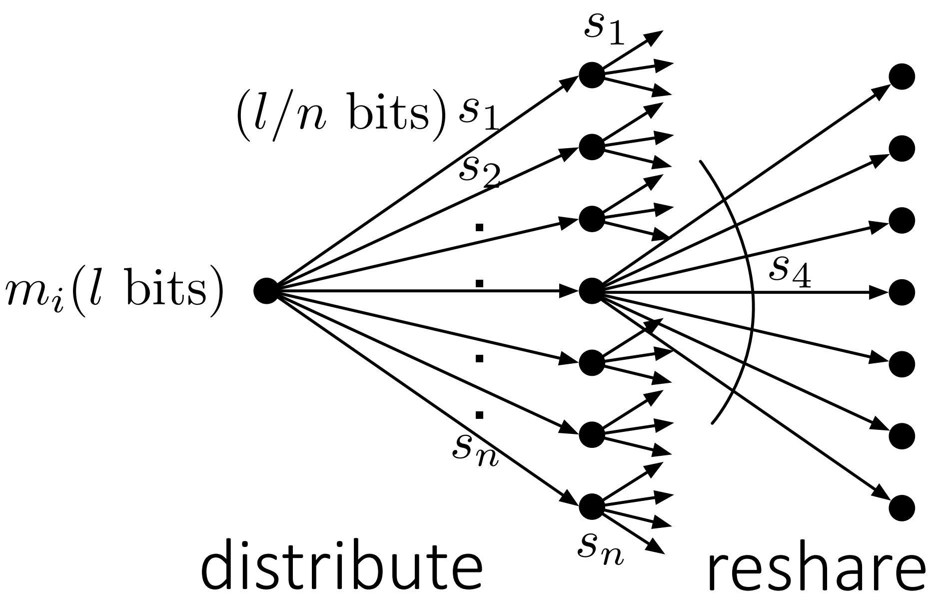


distribute

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

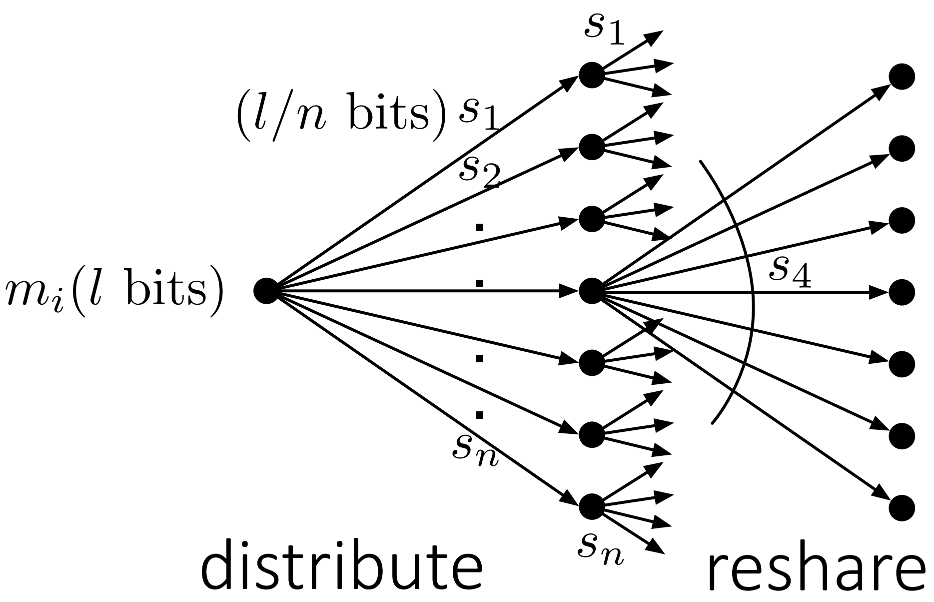
- Agree on a k-bit accumulator value corresponding to one of the inputs, requires O(kn^2) communication
- Share the l-bit input using erasure coding techniques



BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires $O(kn^2)$ communication
- Share the l-bit input using erasure coding techniques

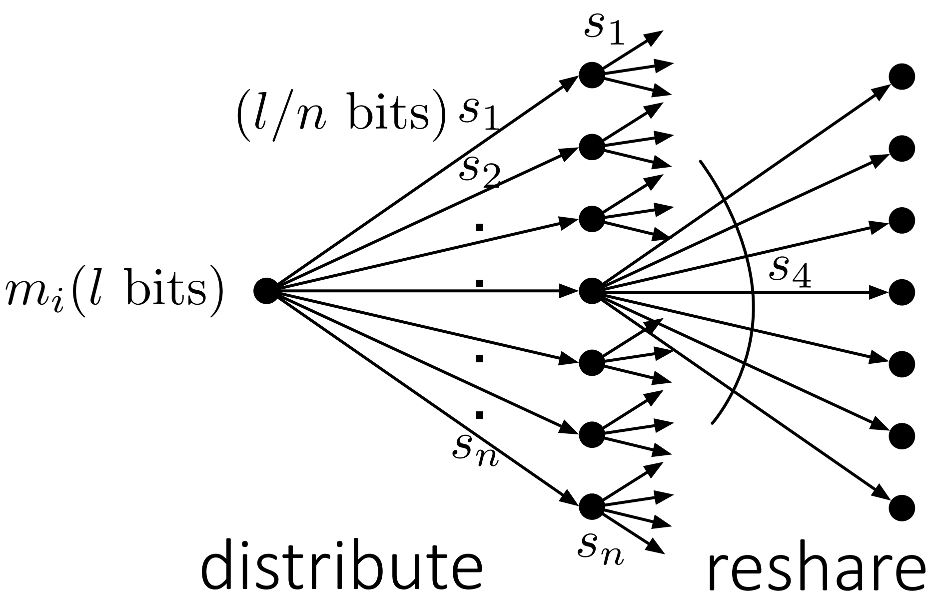


Distribute phase: O(ln)

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires O(kn^2) communication
- Share the l-bit input using erasure coding techniques



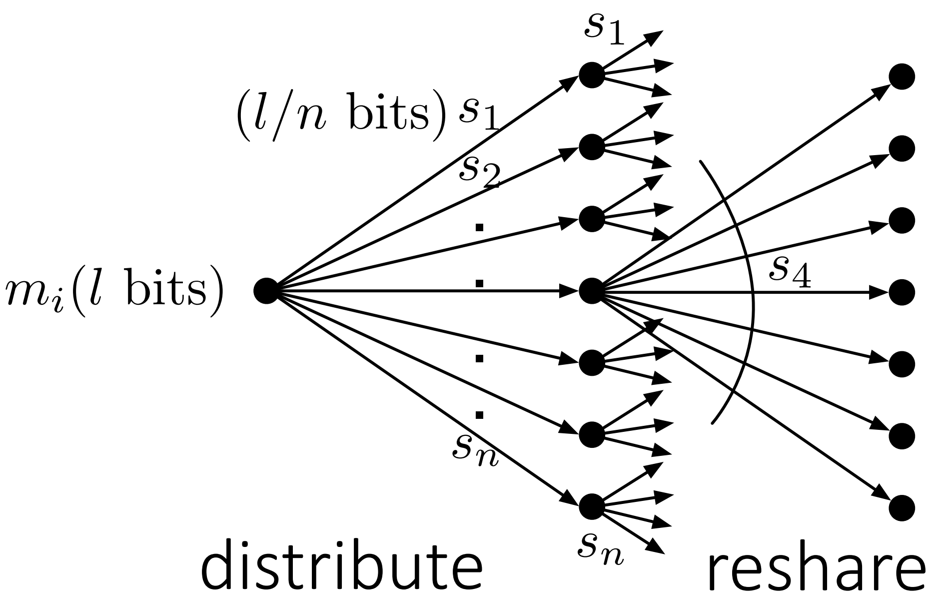
Distribute phase: O(ln)

Reshare phase: O(n^2.l/n) = O(ln)

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires O(kn^2) communication
- Share the l-bit input using erasure coding techniques



Agreeing on accumulator: O(kn^2)

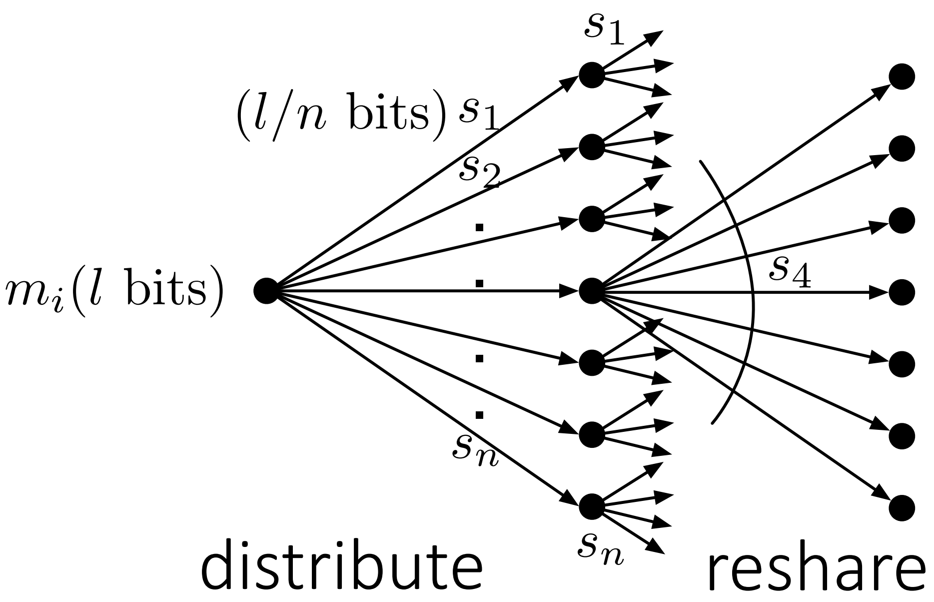
Distribute phase: O(ln)

Reshare phase: O(n^2.l/n) = O(ln)

BA Extension Protocols for l-bit Inputs [NRSVX'20]

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires O(kn²) communication
- Share the l-bit input using erasure coding techniques



Agreeing on accumulator: $O(kn^2)$

Distribute phase: $O(ln)$

Reshare phase: $O(n^2.l/n) = O(ln)$

Total communication: $O(ln+kn^2)$ bits

State of the Art

State of the Art

Sub-quadratic communication complexity against an adaptive adversary: O(poly(k).nl) bits

- Not optimal when l is large

BA Extension protocol for long messages: $O(ln + kn^2)$ bits

- Not optimal when $l < kn$

Can we get the best of both worlds? i.e.,

Can we obtain a communication complexity of $O(ln + poly(k).n)$ bits

Can we get the best of both worlds? i.e.,

Can we obtain a communication complexity of O(ln + poly(k).n) bits under an adaptive adversary?

Attempt 1: Using the [NRSVX'20] Approach

Attempt 1: Using the [NRSVX'20] Approach

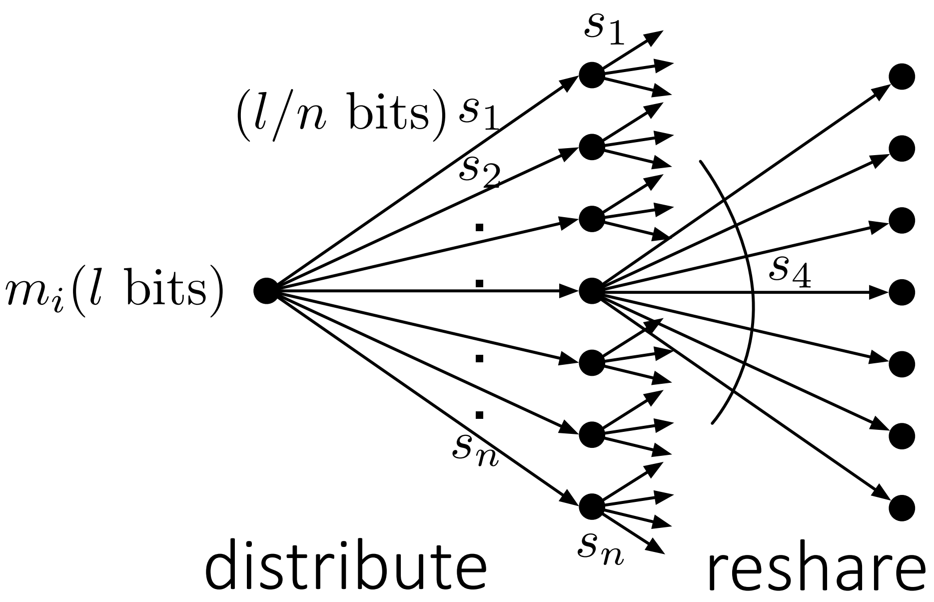
Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires ~~$O(kn^2)$~~ O(poly(k).n) bits of communication
- Share the l-bit input using erasure coding techniques

Attempt 1: Using the [NRSVX'20] Approach

Intuition: Break down the problem into two steps

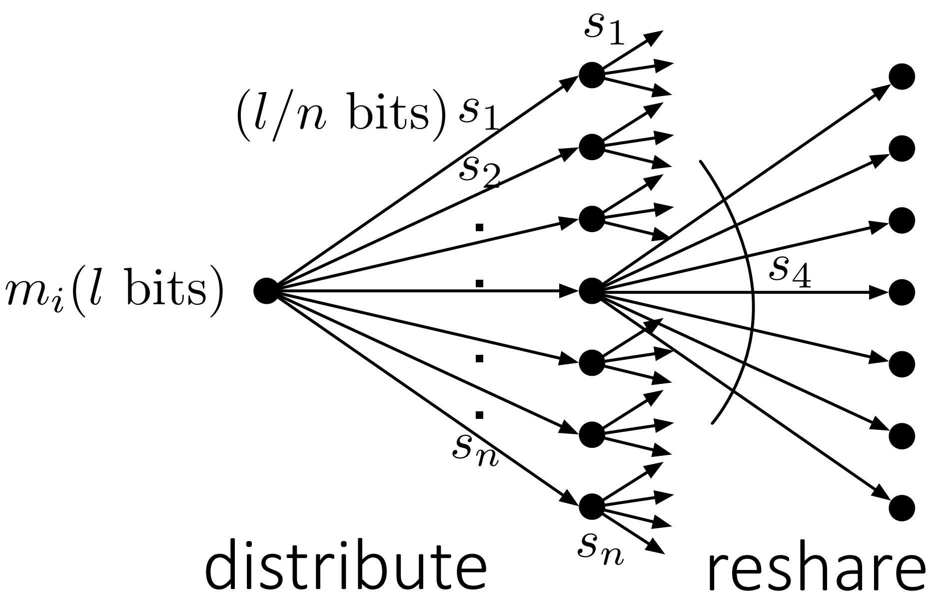
- Agree on a k-bit accumulator value corresponding to one of the inputs, requires ~~$O(kn^2)$~~ O(poly(k).n) bits of communication
- Share the l-bit input using erasure coding techniques



Attempt 1: Using the [NRSVX'20] Approach

Intuition: Break down the problem into two steps

- Agree on a k-bit accumulator value corresponding to one of the inputs, requires ~~$O(kn^2)$~~ O(poly(k).n) bits of communication
- Share the l-bit input using erasure coding techniques



Concern: Even if each party shares 1-bit value in the reshare phase, communication is $\Omega(n^2)$ bits

Attempt 2: Use Multiple k-sized Committees

Attempt 2: Use Multiple k-sized Committees

Requirement: Split the message into k shares and each share of the message should be shared by some honest party

Approach: Use an O(k)-sized committee for resharing each share

Two drawbacks/challenges:

(i) Communication complexity for resharing each share: $\Omega(nk.(l/k))$; for k shares, it is $\Omega(nkl)$

(ii) Adaptivity: How do we distribute these shares with k different committees?

Our Solution: Approach

Our Solution: Approach

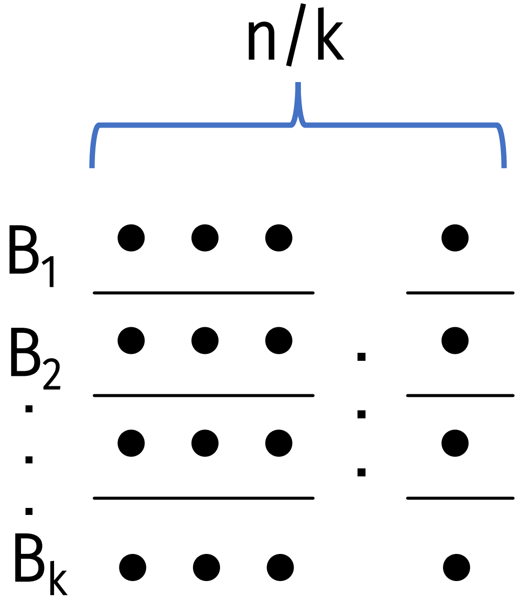
Approach:

- Publicly split the parties into k buckets of size n/k
- Distribute: Share i is shared with parties in bucket i
- Reshare: Elect single O(k)-sized committee; bucket i parties reshare share i

Our Solution: Approach

Approach:

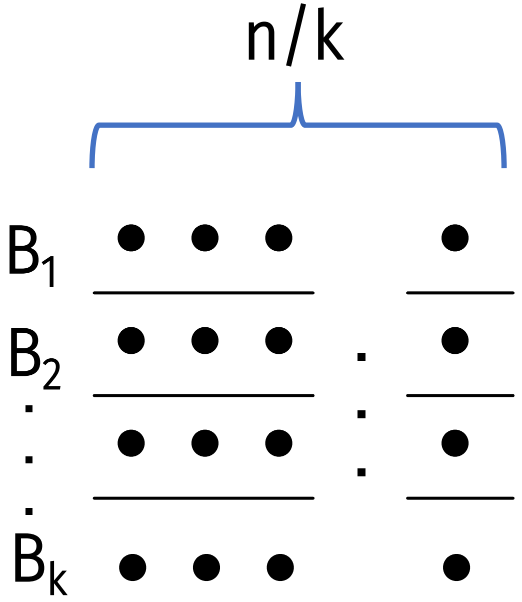
- Publicly split the parties into k buckets of size n/k
- Distribute: Share i is shared with parties in bucket i
- Reshare: Elect single O(k)-sized committee; bucket i parties reshare share i



Our Solution: Approach

Approach:

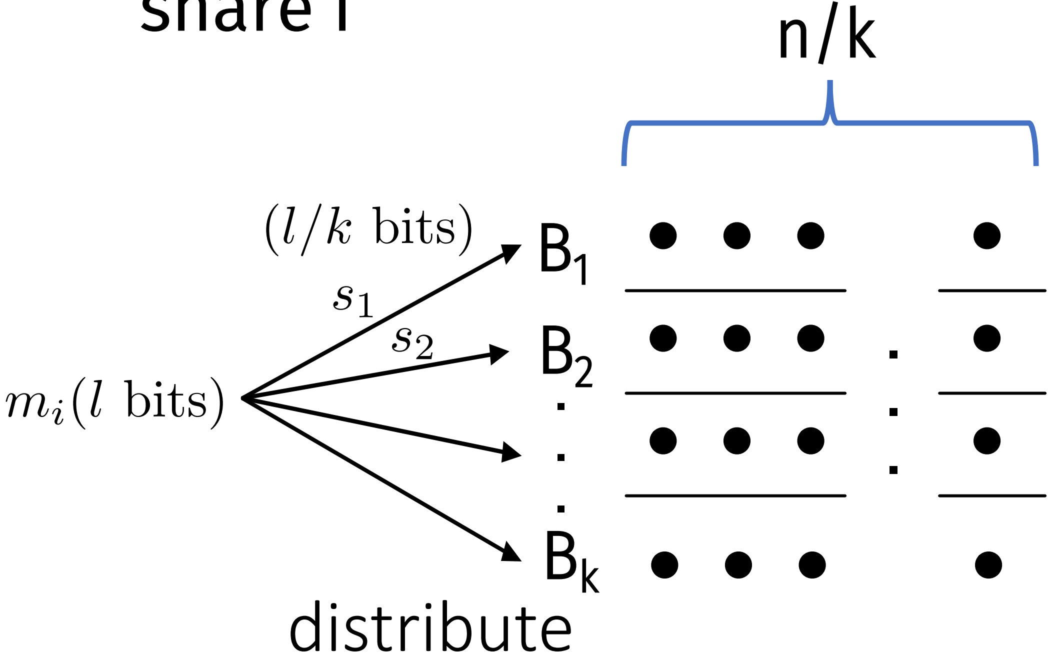
- Publicly split the parties into k buckets of size n/k
- Distribute: Share i is shared with parties in bucket i
- Reshare: Elect single O(k)-sized committee; bucket i parties reshare share i



Our Solution: Approach

Approach:

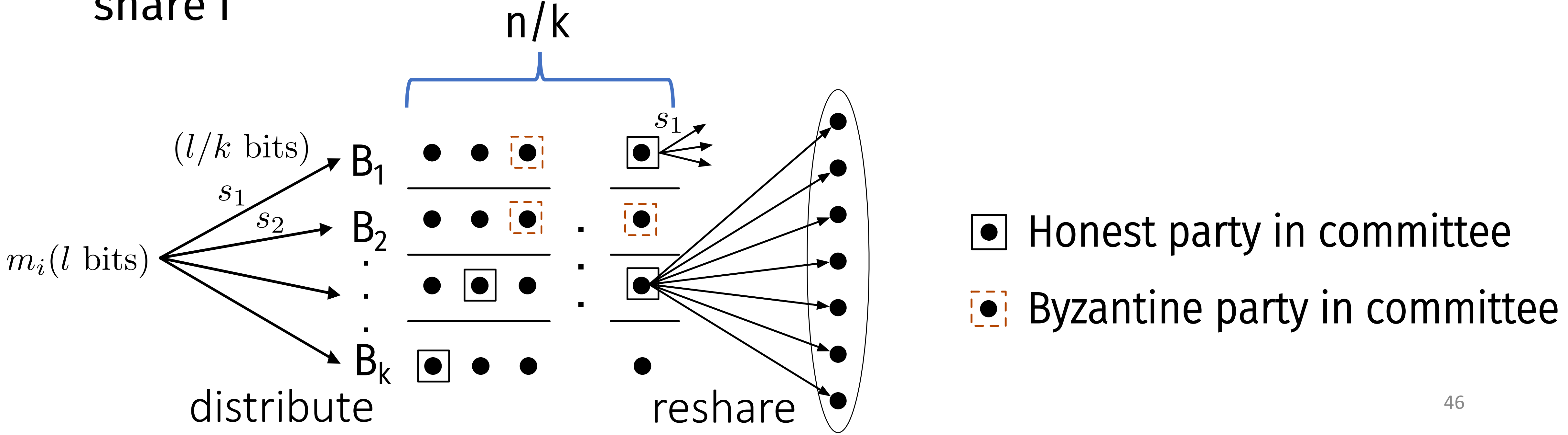
- Publicly split the parties into k buckets of size n/k
- Distribute: Share i is shared with parties in bucket i
- Reshare: Elect single O(k)-sized committee; bucket i parties reshare share i



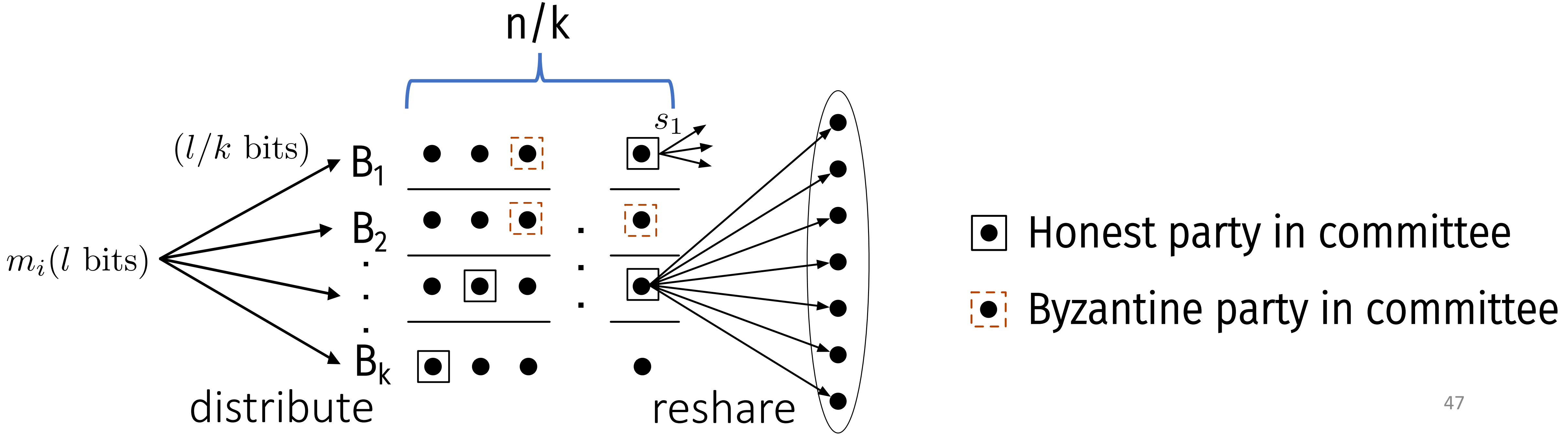
Our Solution: Approach

Approach:

- Publicly split the parties into k buckets of size n/k
- Distribute: Share i is shared with parties in bucket i
- Reshare: Elect single O(k)-sized committee; bucket i parties reshare share i

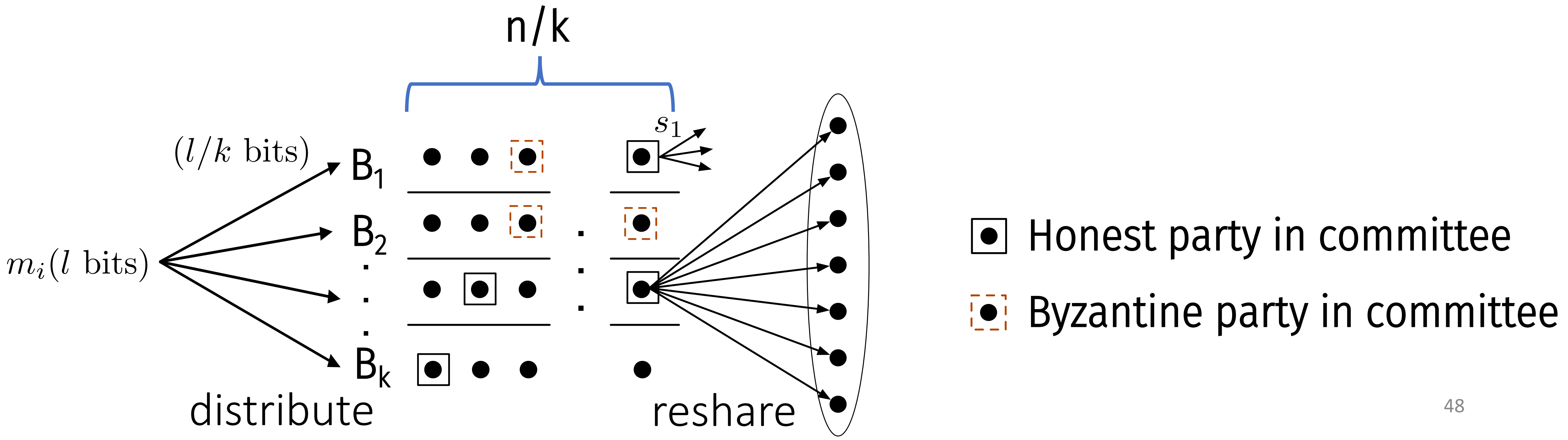


Our Solution: Communication Complexity

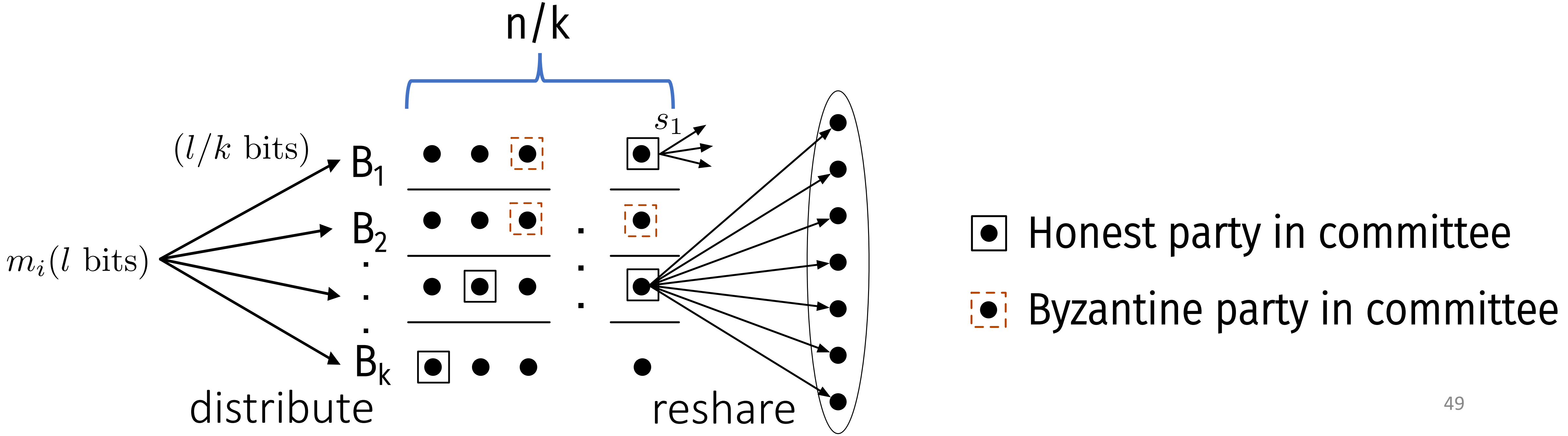


Our Solution: Communication Complexity

- Distribute: O(k) parties sharing l/k-sized shares to n/k parties = O(ln/k) bits per share
- Reshare: O(k) parties sharing l/k-sized shares to n parties = O(ln) bits



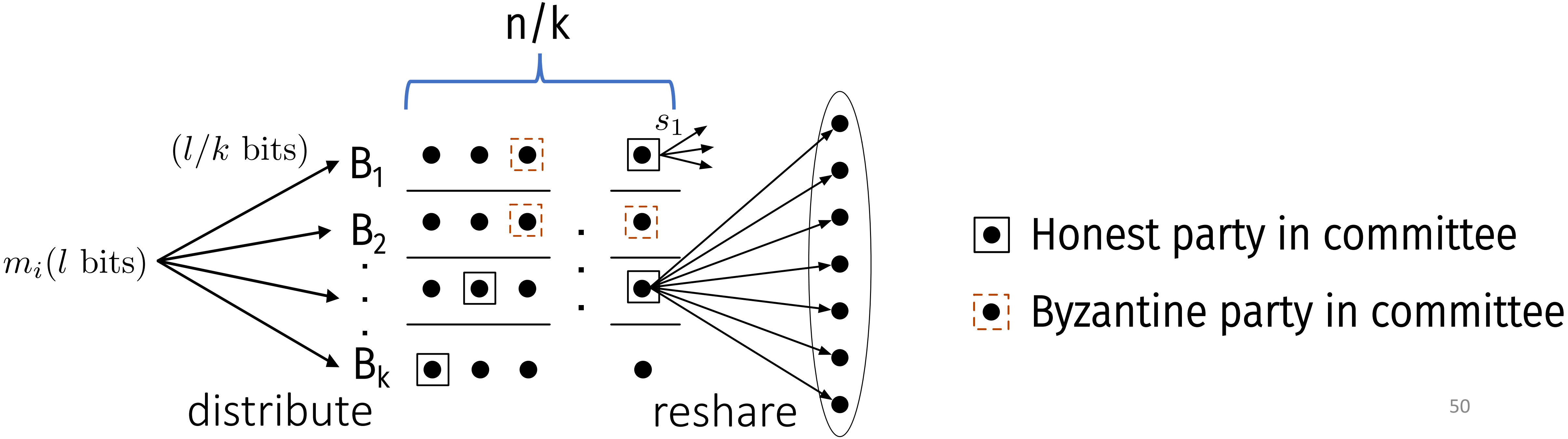
Our Solution: Potential Concerns



Our Solution: Potential Concerns

(i) Are enough shares reshared? Each bucket i has only O(1) parties who can reshare share i

(ii) Adaptivity: The adversary can adaptively corrupt parties in different public buckets so that not enough shares are reshared



(i) Are enough shares reshared? Each bucket i has only O(1) parties who can reshare share i

(ii) Adaptivity: The adversary can adaptively corrupt parties in different public buckets so that not enough shares are reshared

- We cannot use Chernoff-type bounds

Solution: A balls-and-bins analysis using McDiarmid's inequality

Our Solution: Analysis using McDiarmid's Inequality

(i) Are enough shares reshared? Each bucket i has only O(1) parties who can reshare share i

(ii) Adaptivity: The adversary can adaptively corrupt parties in different public buckets so that not enough shares are reshared

- We cannot use Chernoff-type bounds

Solution: A balls-and-bins analysis using McDiarmid's inequality

Our Result

Our Result

Theorem: For any $\varepsilon > 0$, assuming appropriate cryptographic assumptions, there exists an adaptively secure BA protocol achieving a communication complexity of $O(nl + poly(k).n)$ for l-bit inputs for

(i) $t < (1-\varepsilon)\, n/2$ Byzantine parties under a synchronous network,

(ii) $t < (1-\varepsilon)\, n/3$ Byzantine parties under an asynchronous network

Thank you!

kartik@cs.duke.edu

Our Result

Theorem: For any $\varepsilon > 0$, assuming appropriate cryptographic assumptions, there exists an adaptively secure BA protocol achieving a communication complexity of $O(nl + poly(k).n)$ for l-bit inputs for

(i) $t < (1- \varepsilon)\, n/2$ Byzantine parties under a synchronous network,

(ii) $t < (1- \varepsilon)\, n/3$ Byzantine parties under an asynchronous network

Thank you!
kartik@cs.duke.edu

https://eprint.iacr.org/2021/1403.pdf