Gossiping for Communication-Efficient Broadcast

G. Tsimos, Julian Loss, Charalampos Papamanthou

UNIVERSITY OF MARYLAND 1856

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Broadcast

Broadcast

- A designated sender s w. input value u_s

Broadcast

- A designated sender s w. input value u_s
- s wants to broadcast u_s to all n parties

Broadcast

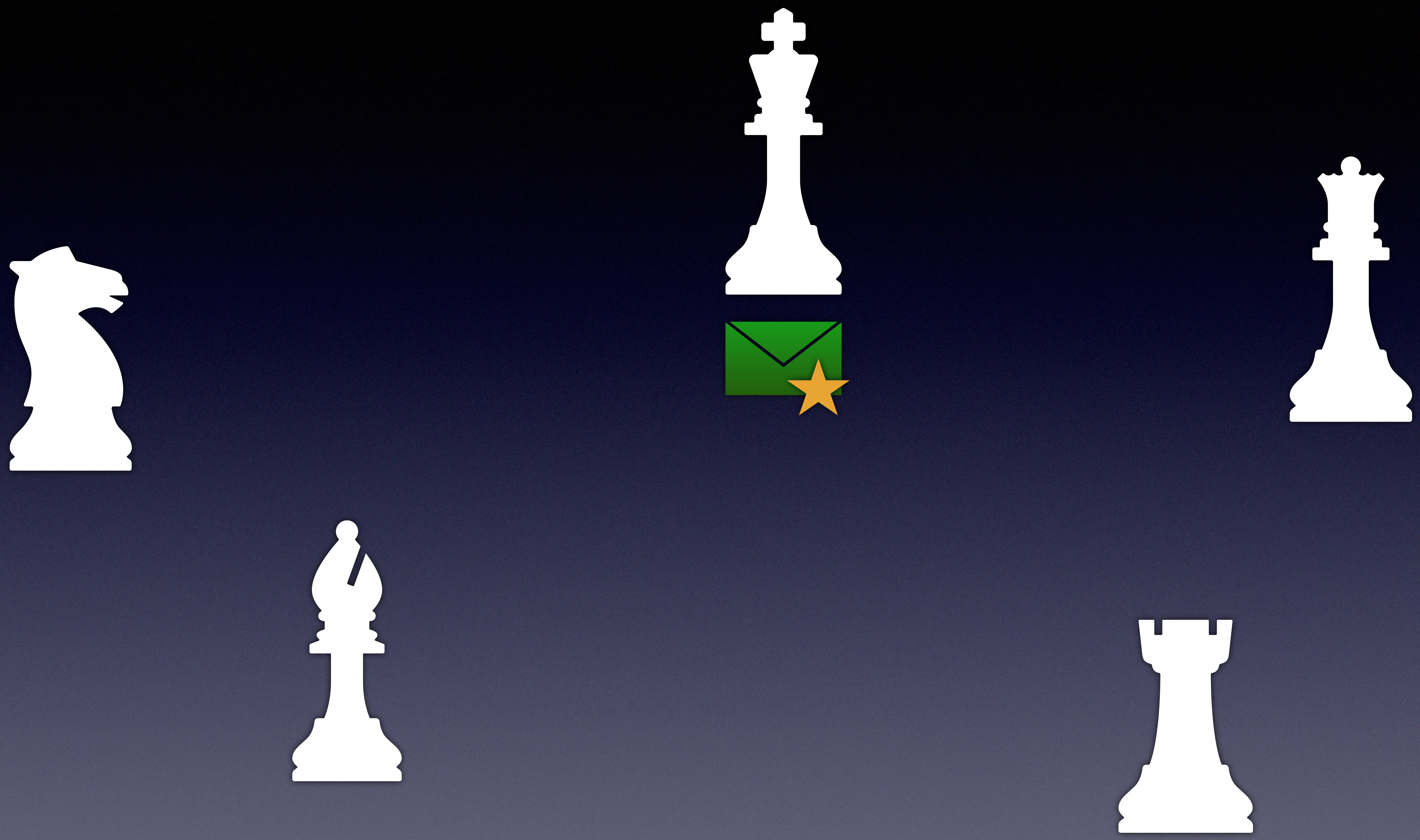
- A designated sender s w. input value u_s
- s wants to broadcast u_s to all n parties
  - s might be dishonest

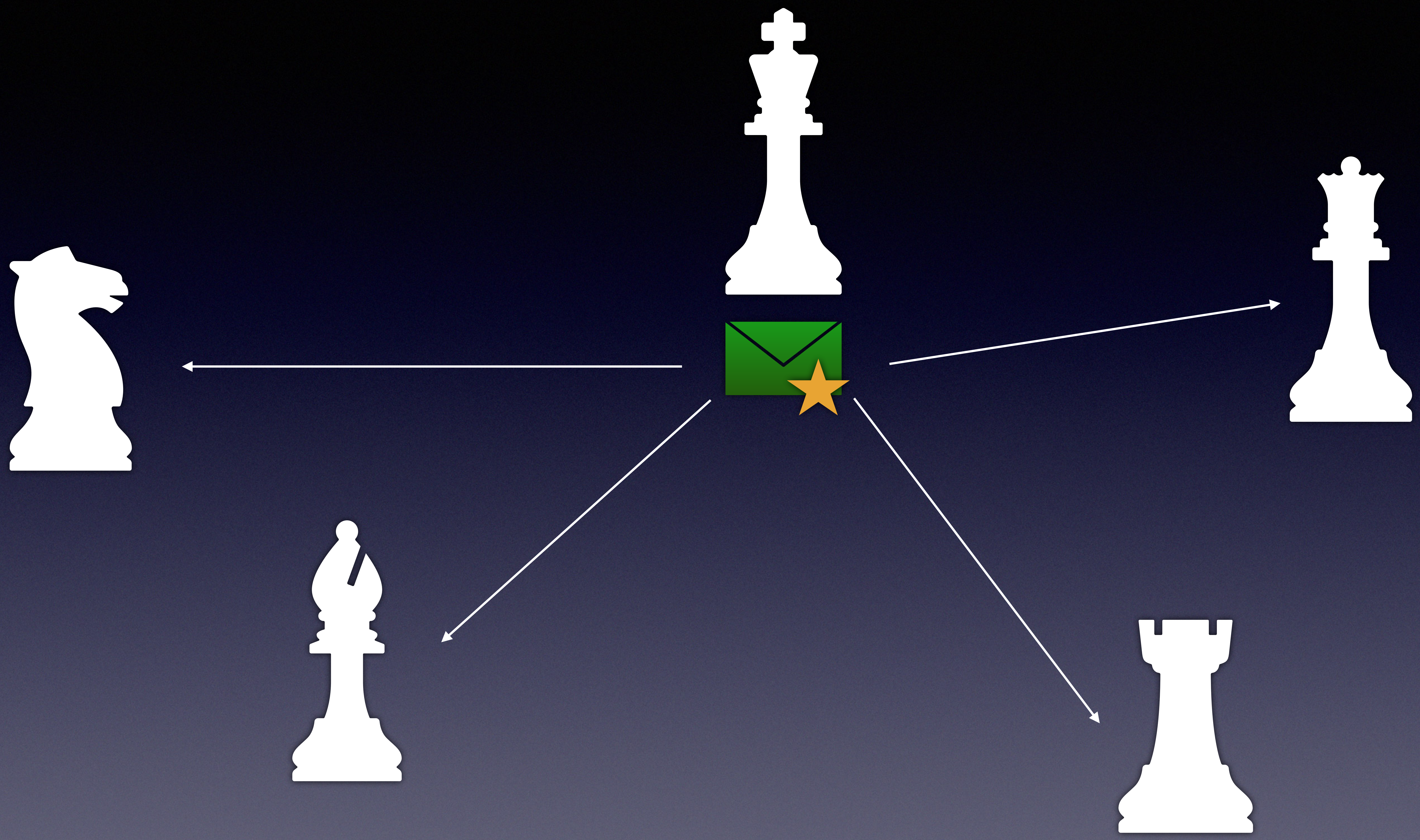
Broadcast

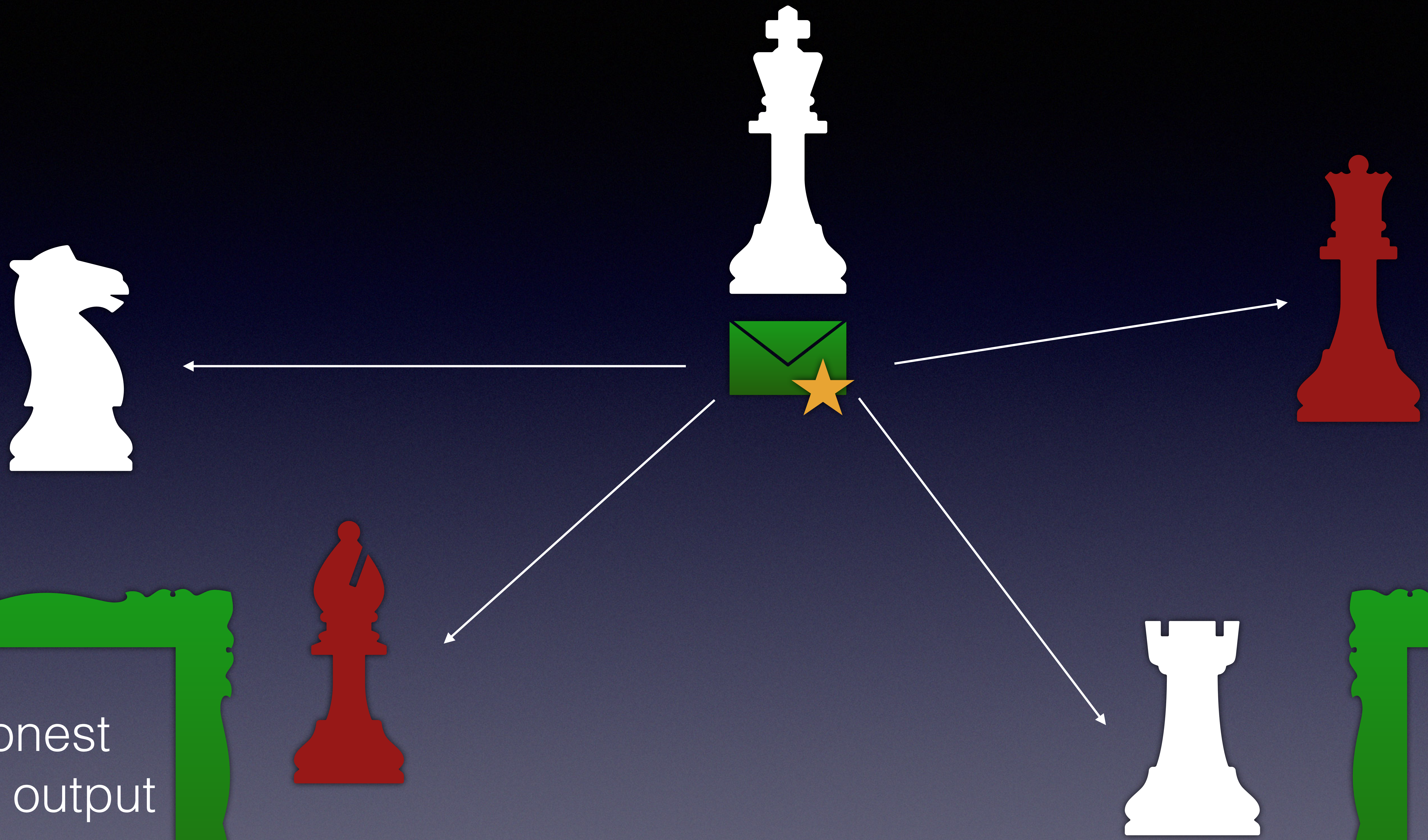
- A designated sender s w. input value u_s
- s wants to broadcast u_s to all n parties
  - s might be dishonest
- Honest parties want to agree on the same value.

Authenticated Broadcast

- Authenticated Broadcast:
  - Broadcast **BUT** with Use of a Public Key Infrastructure (PKI)
    - Bulletin Board \ Trusted PKI
  - Each party can sign with a **signature** each message they send
    - P_i holds (pk_i, sk_i) and posts pk_i publicly







All honest parties output the same message

If S is honest, all honest parties output S's message

Authenticated Broadcast

Authenticated Broadcast

- Multiple settings depending on:

Authenticated Broadcast

- Multiple settings depending on:
  - Synchronous/Asynchronous Communication

Authenticated Broadcast

- Multiple settings depending on:
  - Synchronous/Asynchronous Communication
  - Number of corruptions (Honest/dishonest majority)

Authenticated Broadcast

- Multiple settings depending on:
  - Synchronous/Asynchronous Communication
  - Number of corruptions (Honest/dishonest majority)
  - Setup assumptions

Authenticated Broadcast

- Multiple settings depending on:
  - Synchronous/Asynchronous Communication
  - Number of corruptions (Honest/dishonest majority)
  - Setup assumptions
  - static/adaptive Adversary

Metrics:

Metrics:

- Communication Complexity (CC)

Metrics:

- Communication Complexity (CC)
  - Amount of bits shared by honest parties

Metrics:

- Communication Complexity (CC)
  - Amount of bits shared by honest parties
- Round Complexity (RC)

Metrics:

- Communication Complexity (CC)
  - Amount of bits shared by honest parties
- Round Complexity (RC)
  - Total number of rounds until termination

Setting

Setting

- Synchronous Communication

Setting

- Synchronous Communication
- Dishonest majority

Setting

- Synchronous Communication
- Dishonest majority
- State-of-the-art (without trusted setup):

Setting

- Synchronous Communication
- Dishonest majority
- State-of-the-art (without trusted setup):
  - Dolev-Strong protocol [DS'83] with $O(n^3)$ Communication

In this work we achieve:

In this work we achieve:

Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions.

In this work we achieve:

Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions.

We introduce **gossiping** and **Converge** and show Parallel Broadcast with:

In this work we achieve:

Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions.

We introduce **gossiping** and Converge and show Parallel Broadcast with:

$\mathcal{O}(n^3)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions.

In this work we achieve:

Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1 - \epsilon)n$ **static** corruptions.

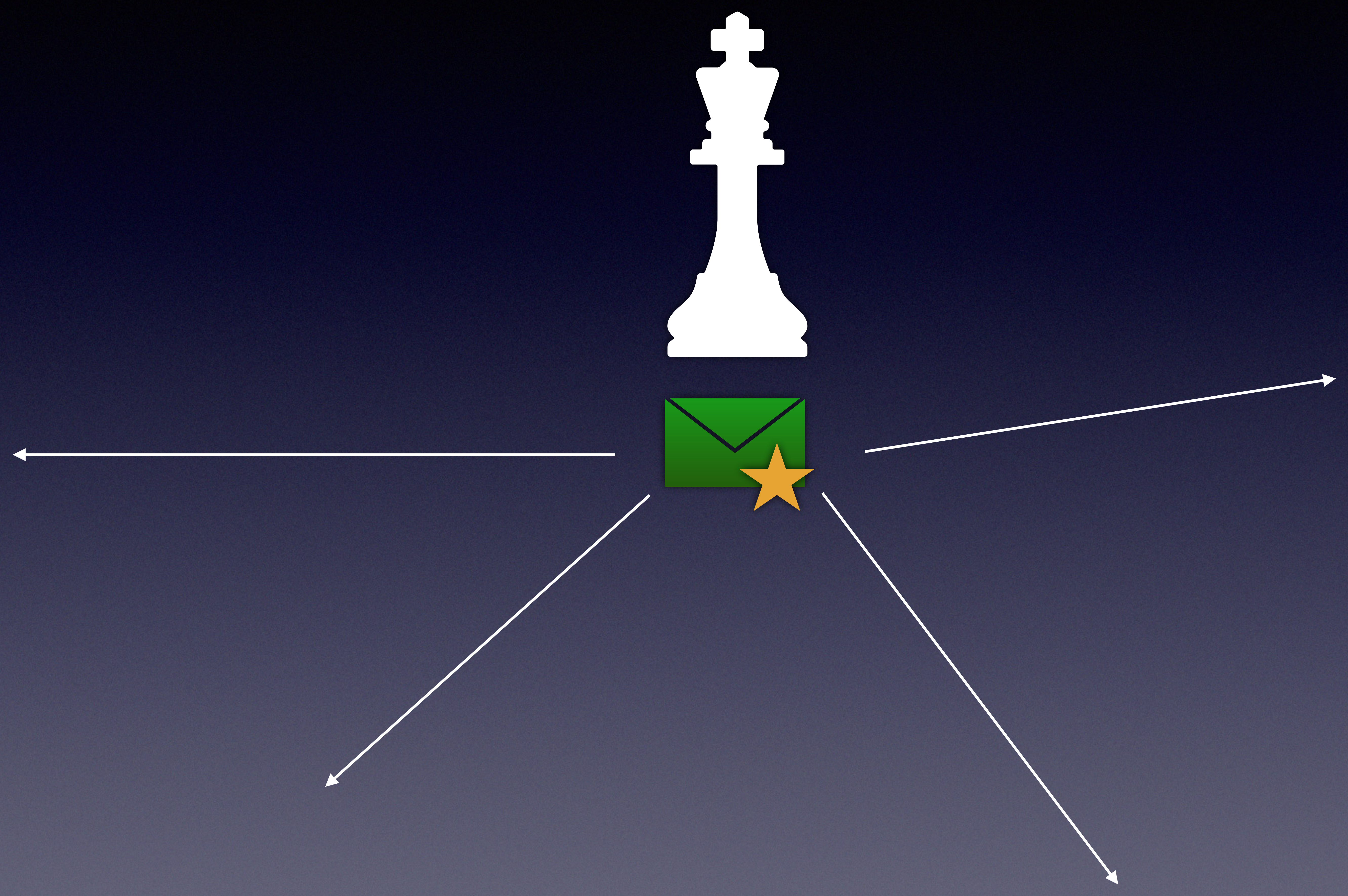
We introduce **gossiping** and Converge and show Parallel Broadcast with:

$\mathcal{O}(n^3)$ CC using **bulletin board PKI**, against $t < (1 - \epsilon)n$ **adaptive** corruptions.

$\mathcal{O}(n^2)$ CC using **trusted PKI**, against $t < (1 - \epsilon)n$ **adaptive** corruptions.

Dolev-Strong

S sends ✉ with S's signature ★ to all parties



For each $r \leq t + 1$:

p checks if it received some new “valid” 

For each $r \leq t + 1$:

p checks if it received some new "valid" ✉

Valid ✉ at round r:

At least r ★ from distinct parties. One is from S.

For each $r \leq t + 1$:

p checks if it received some new “valid”

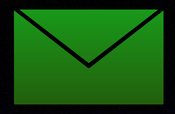
If so, it adds its signature and sends … to all parties.

For each $r \leq t + 1$:

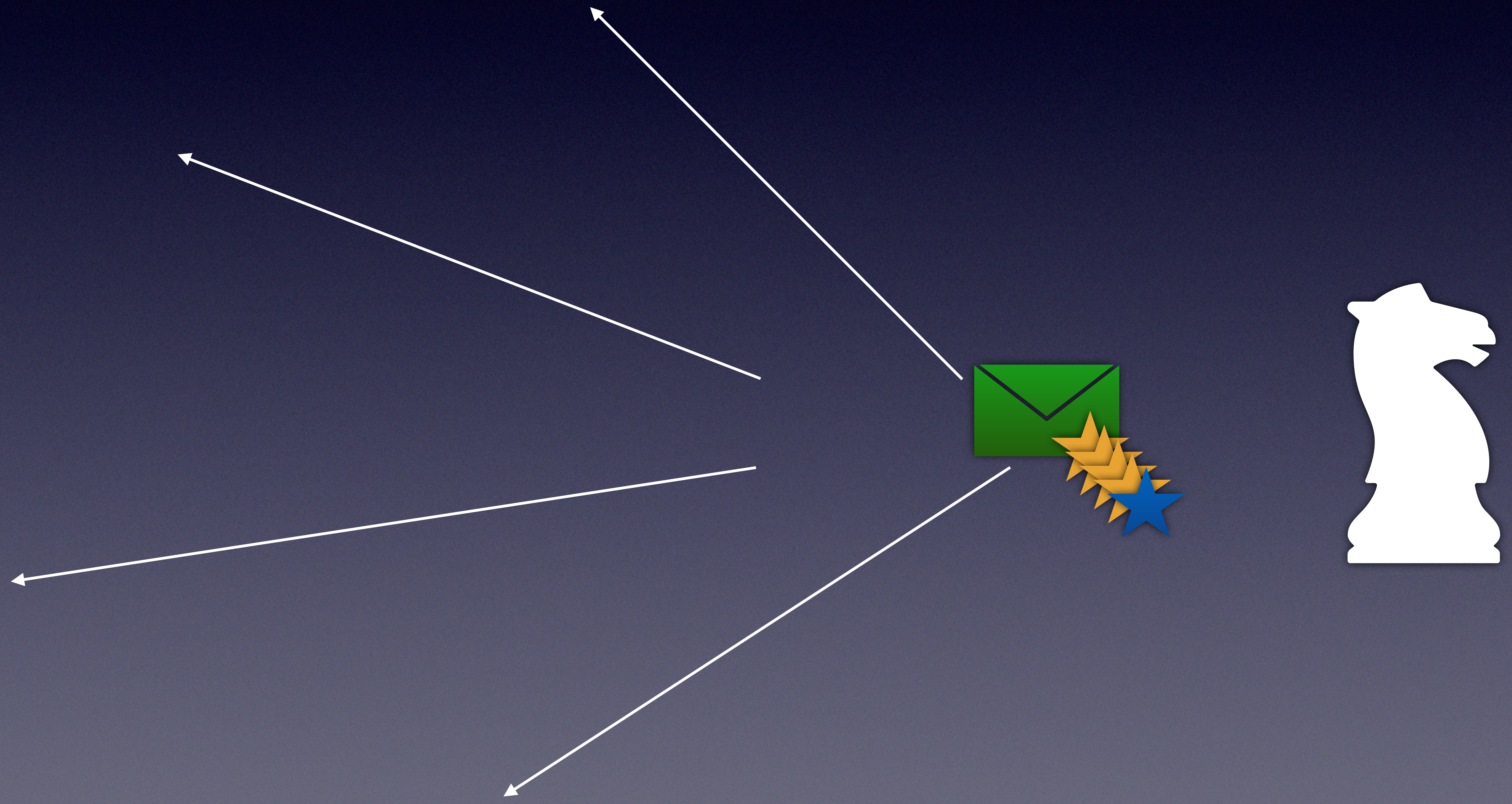
p checks if it received some new “valid” ✉

If so, it adds its signature ★ and sends ✉★…★★ to all parties.

For each $r \leq t + 1$:

p checks if it received some new "valid" 

If so, it adds its signature  and sends  to all parties.



Dolev-Strong Protocol

Dolev-Strong Protocol

- Achieves Authenticated Broadcast:

Dolev-Strong Protocol

- Achieves Authenticated Broadcast:
  - For any $t < n$ adaptive corruptions

Dolev-Strong Protocol

- Achieves Authenticated Broadcast:
  - For any $t < n$ adaptive corruptions
  - Deterministic, $t + 1 = \mathcal{O}(n)$ rounds

Dolev-Strong Protocol

- Achieves Authenticated Broadcast:
  - For any $t < n$ adaptive corruptions
  - Deterministic, $t + 1 = \mathcal{O}(n)$ rounds
  - Assumes only bulletin board PKI

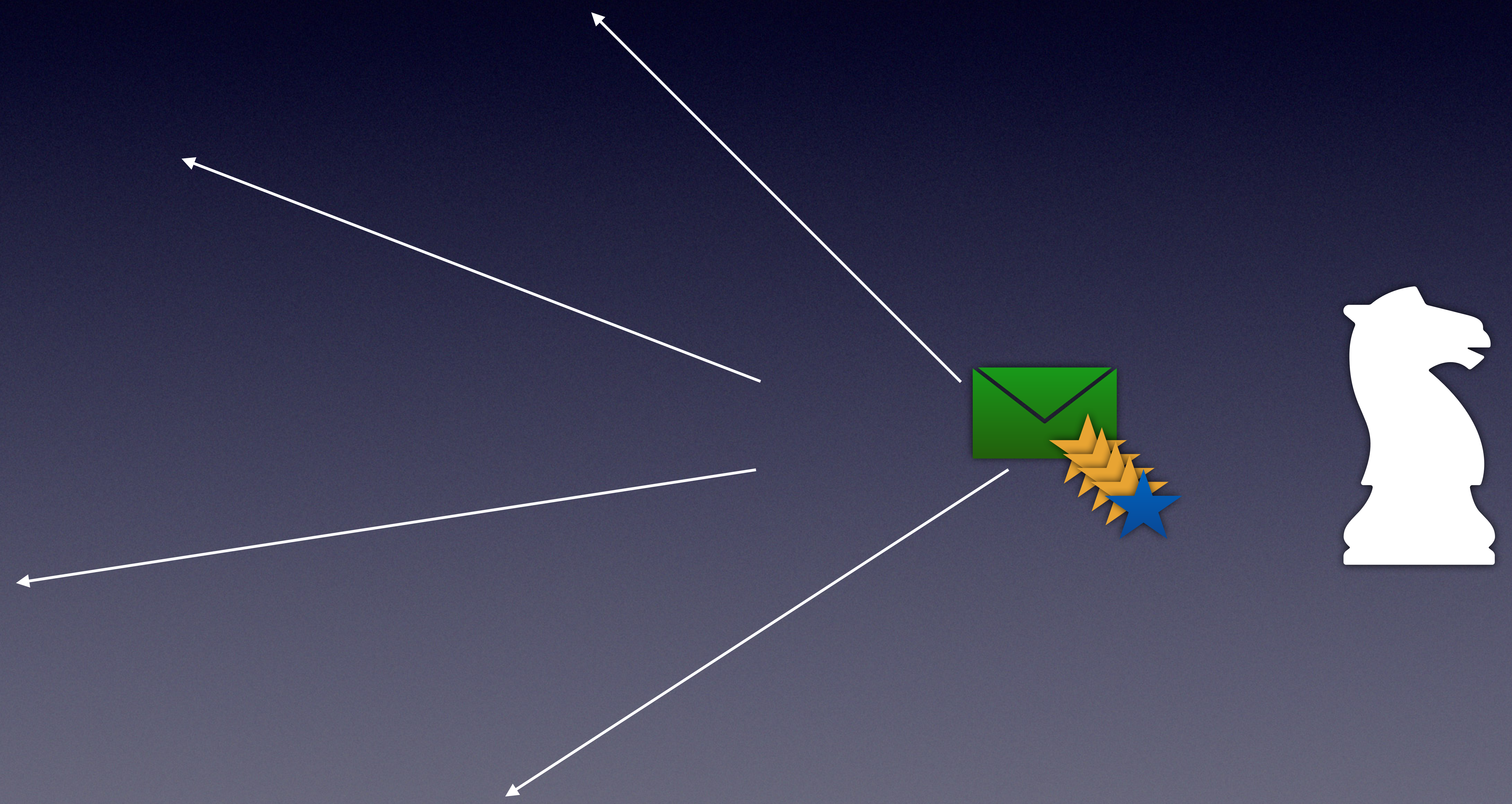
Dolev-Strong Protocol

- Achieves Authenticated Broadcast:
  - For any $t < n$ adaptive corruptions
  - Deterministic, $t + 1 = \mathcal{O}(n)$ rounds
  - Assumes only bulletin board PKI
  - With $\mathcal{O}(n^3\kappa)$ Communication

For each $r \leq t+1$:

p checks if it received some new “valid” ✉

If so, it adds its signature ★ and sends ✉★…★★ to all parties.



Our Observation

- What do parties want to achieve with sending?
  - Perhaps, sending to everyone takes more communication than what needed for the property.

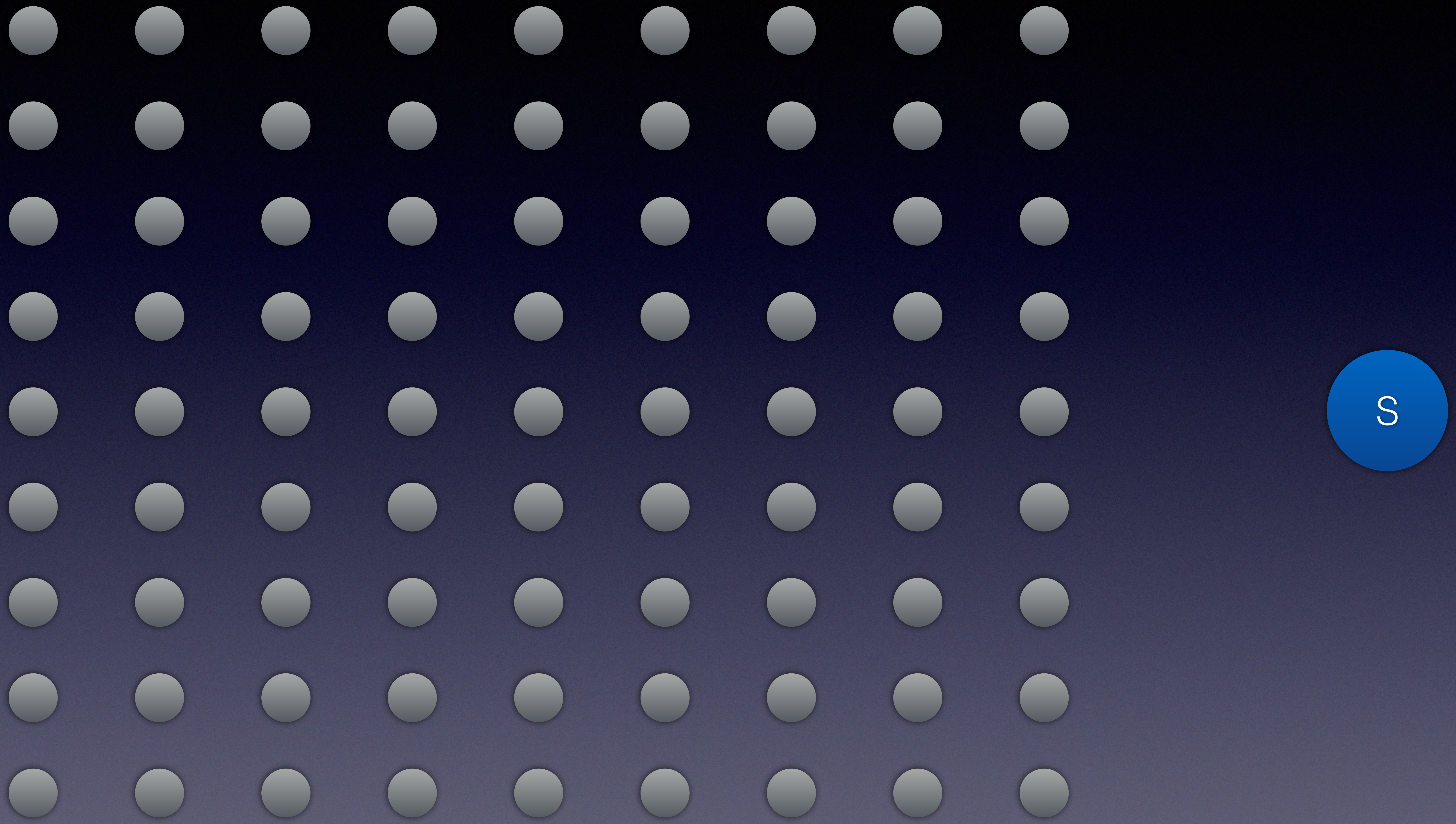

"Do I send the message to party j?"

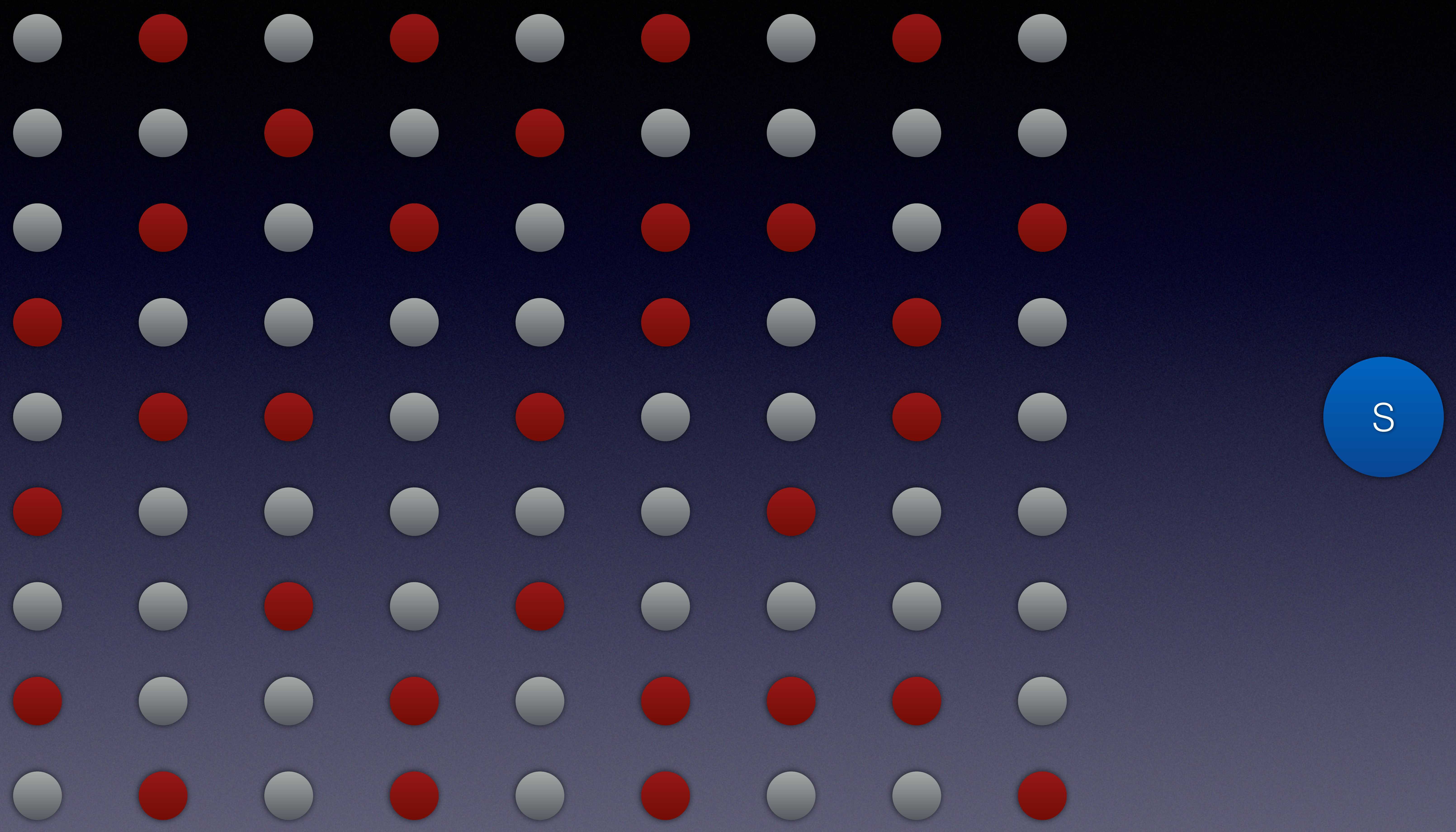
“Do I send the message to party j?”

Flip a coin with prob. m/n
If Heads, then I send, else I don’t

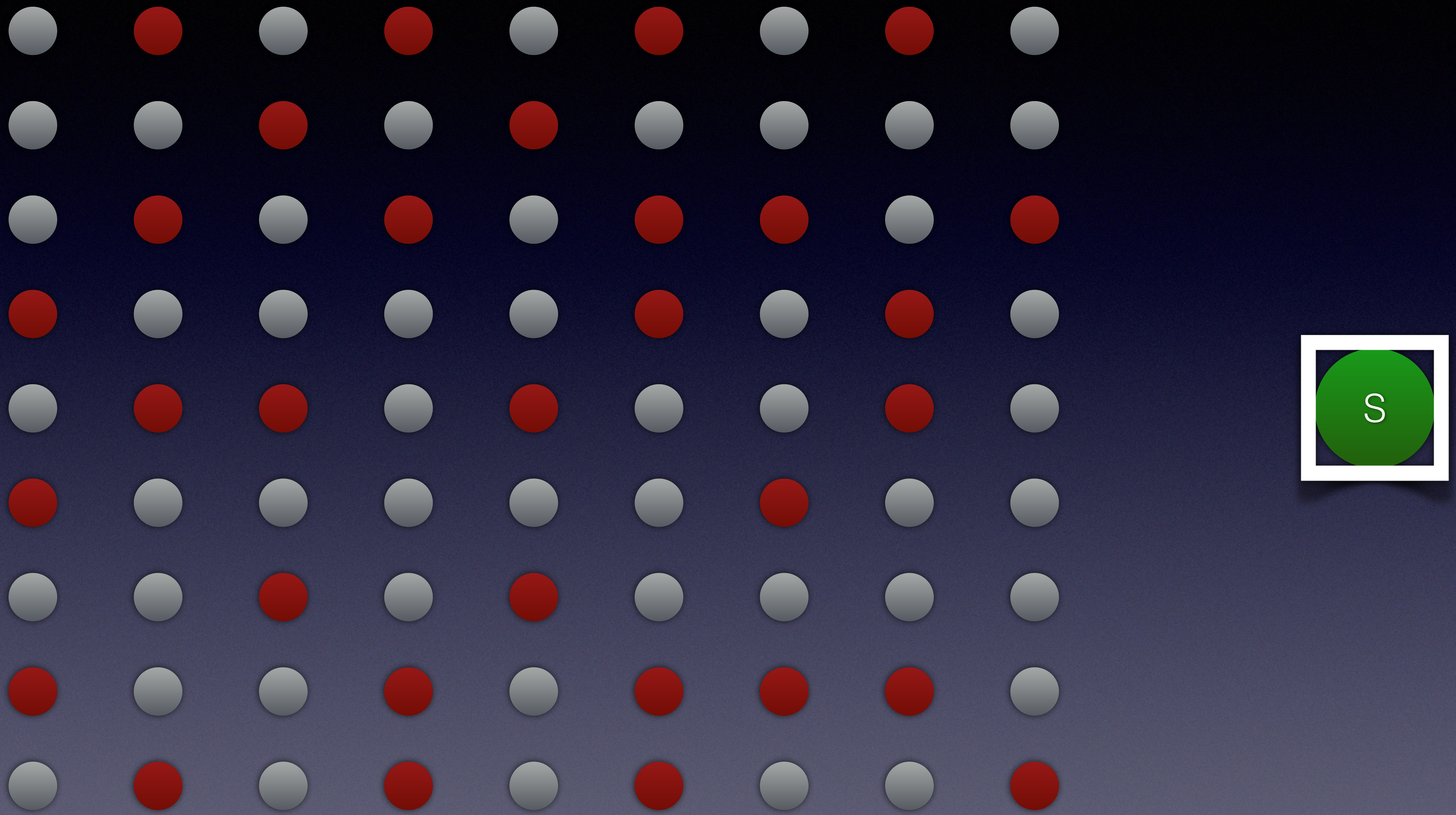
Our Idea for BC

- Gossiping:
  - Each honest party picks randomly ~$\mathcal{O}(\log n)$ other parties to send to.
  - (Ofc, this doesn't work single-shot.) Takes ~$\mathcal{O}(\log n)$ rounds.

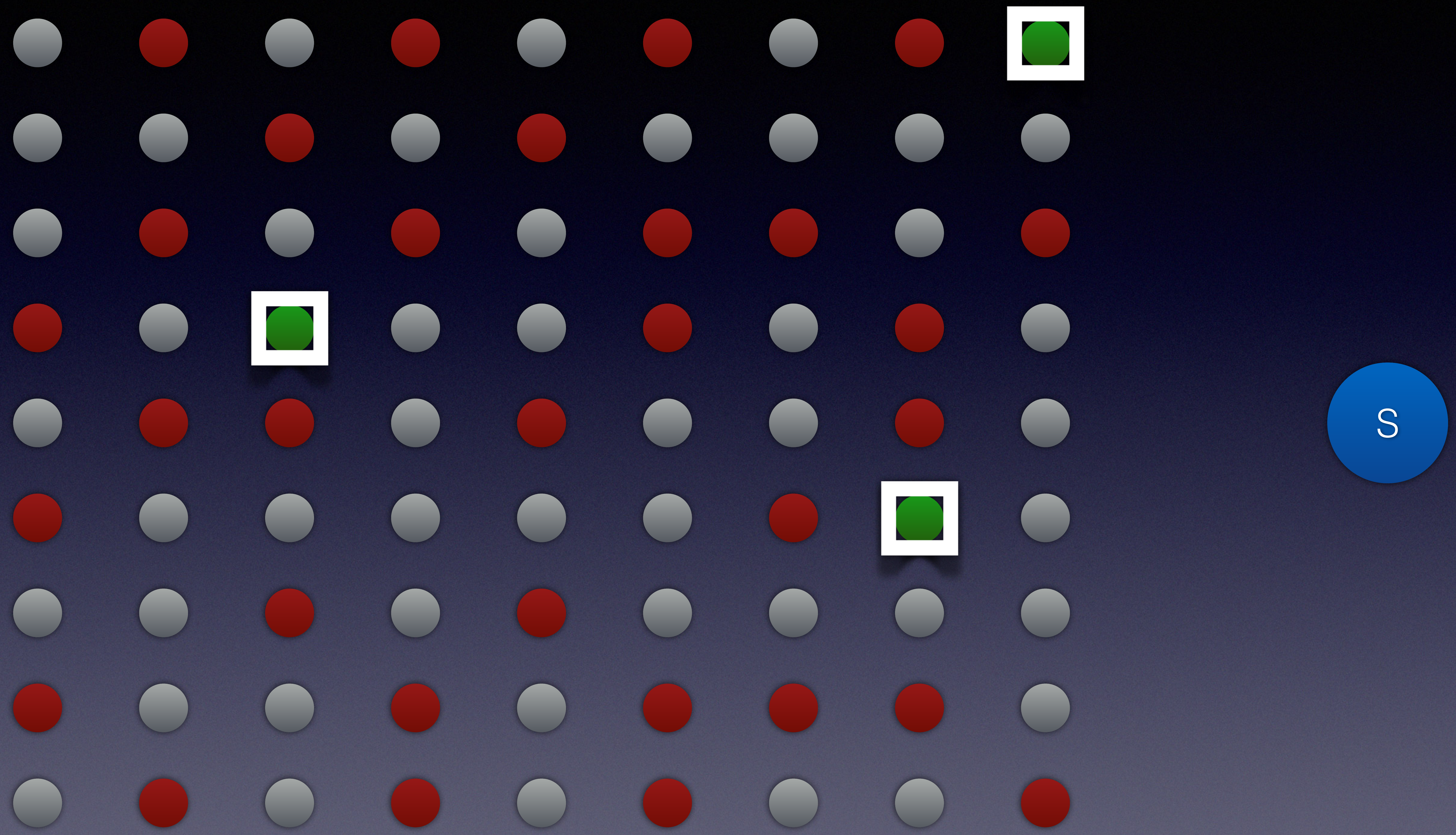

S


S

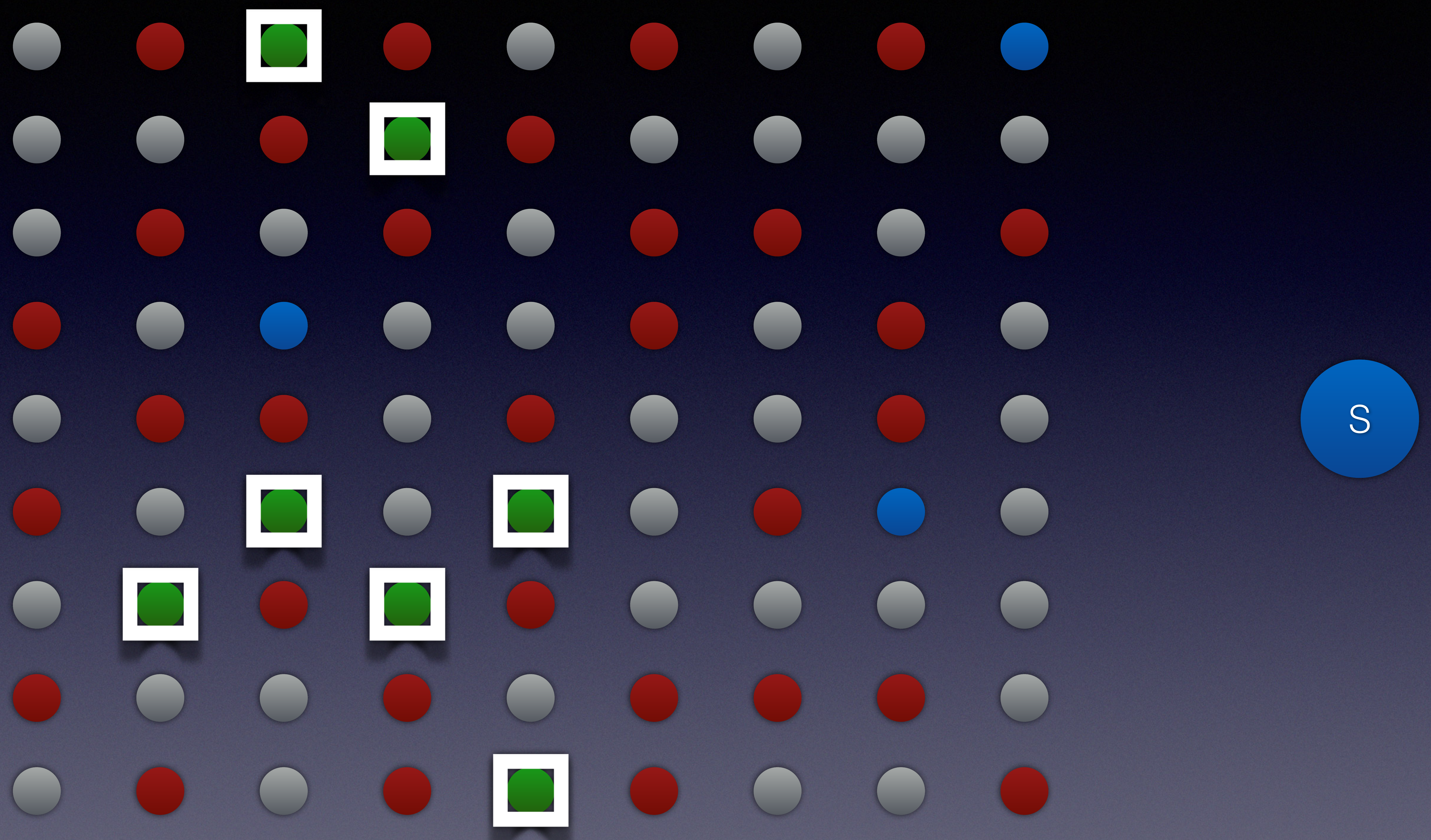
r=a<t+1



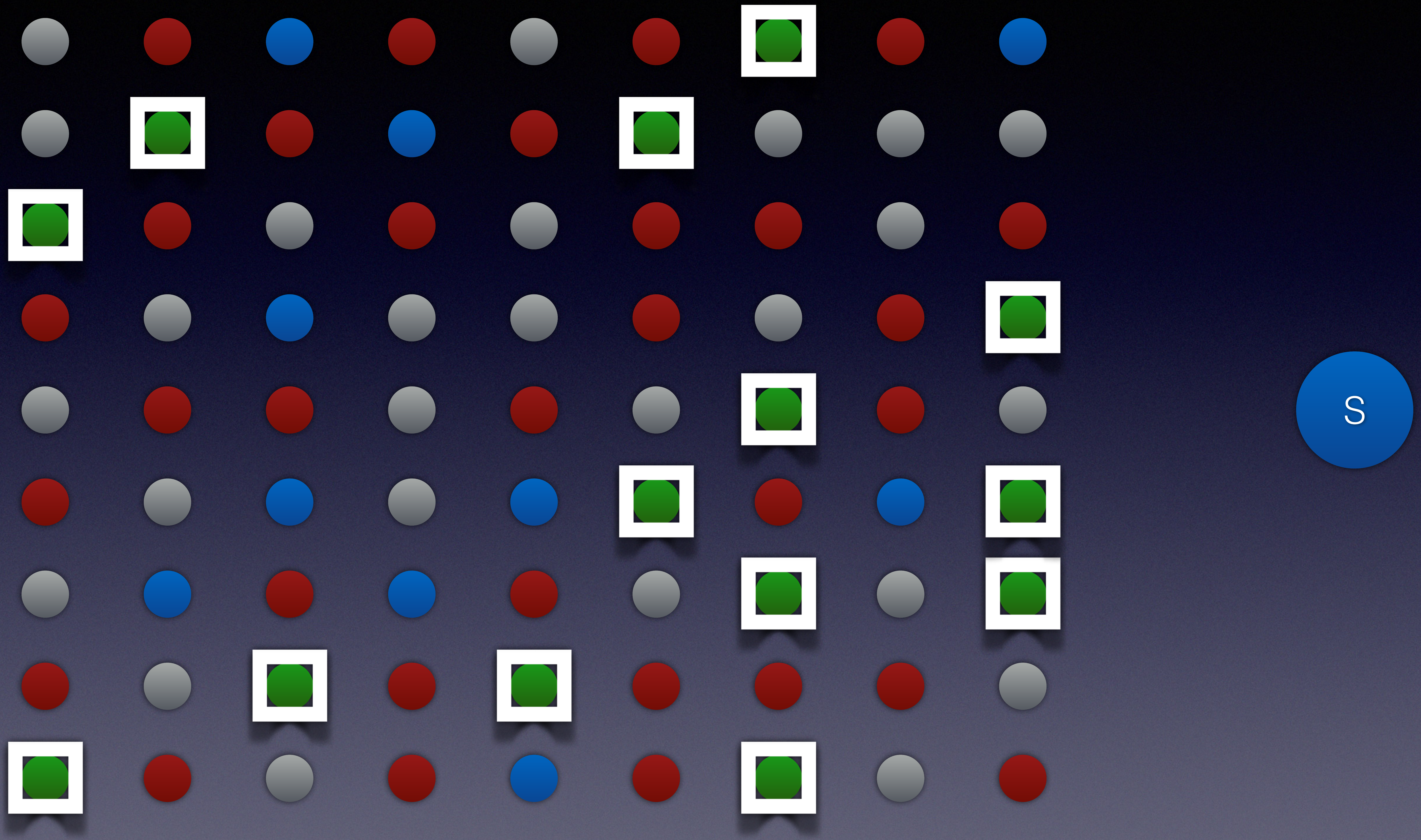
$r=a+1$



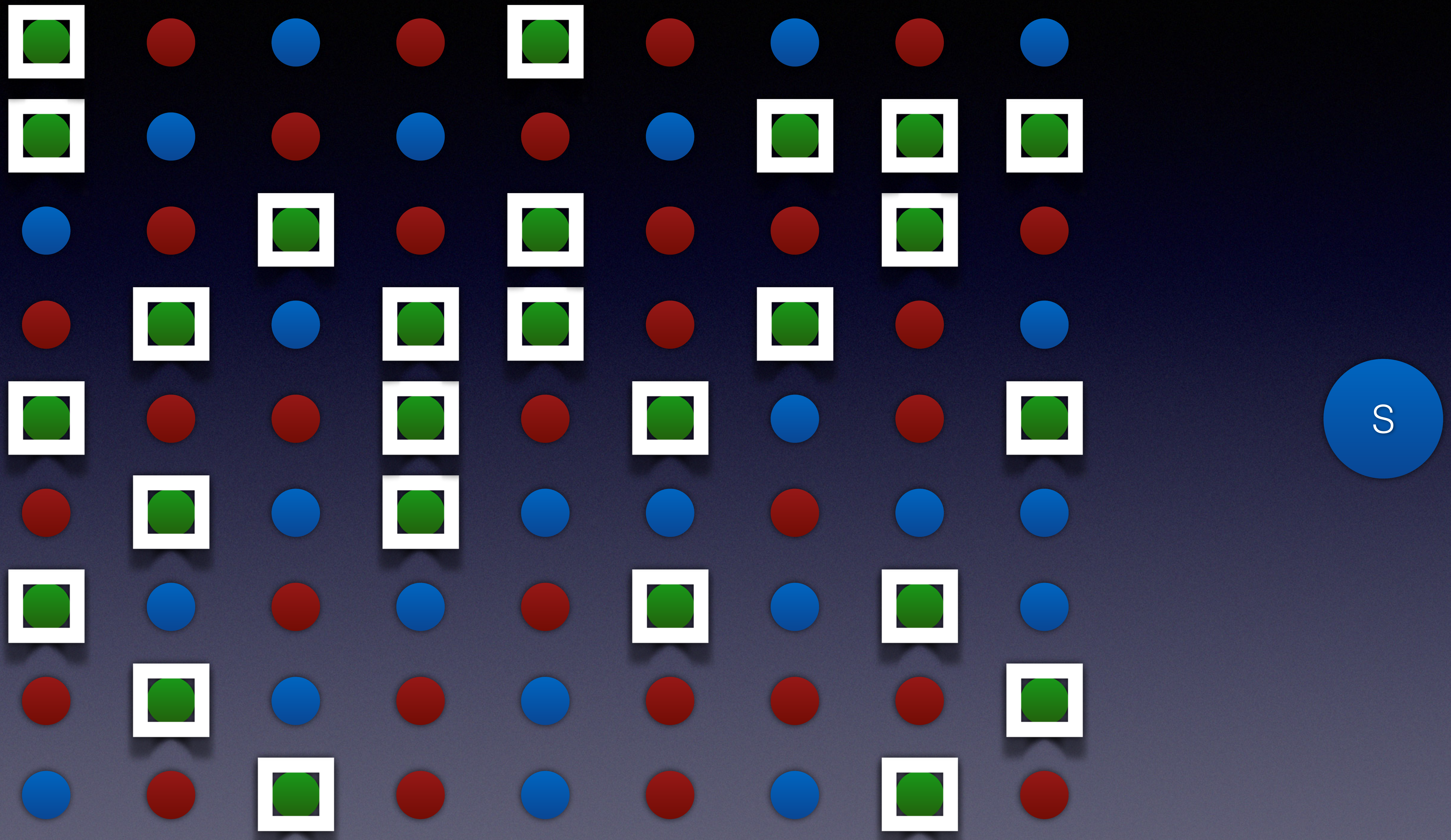
r=a+2



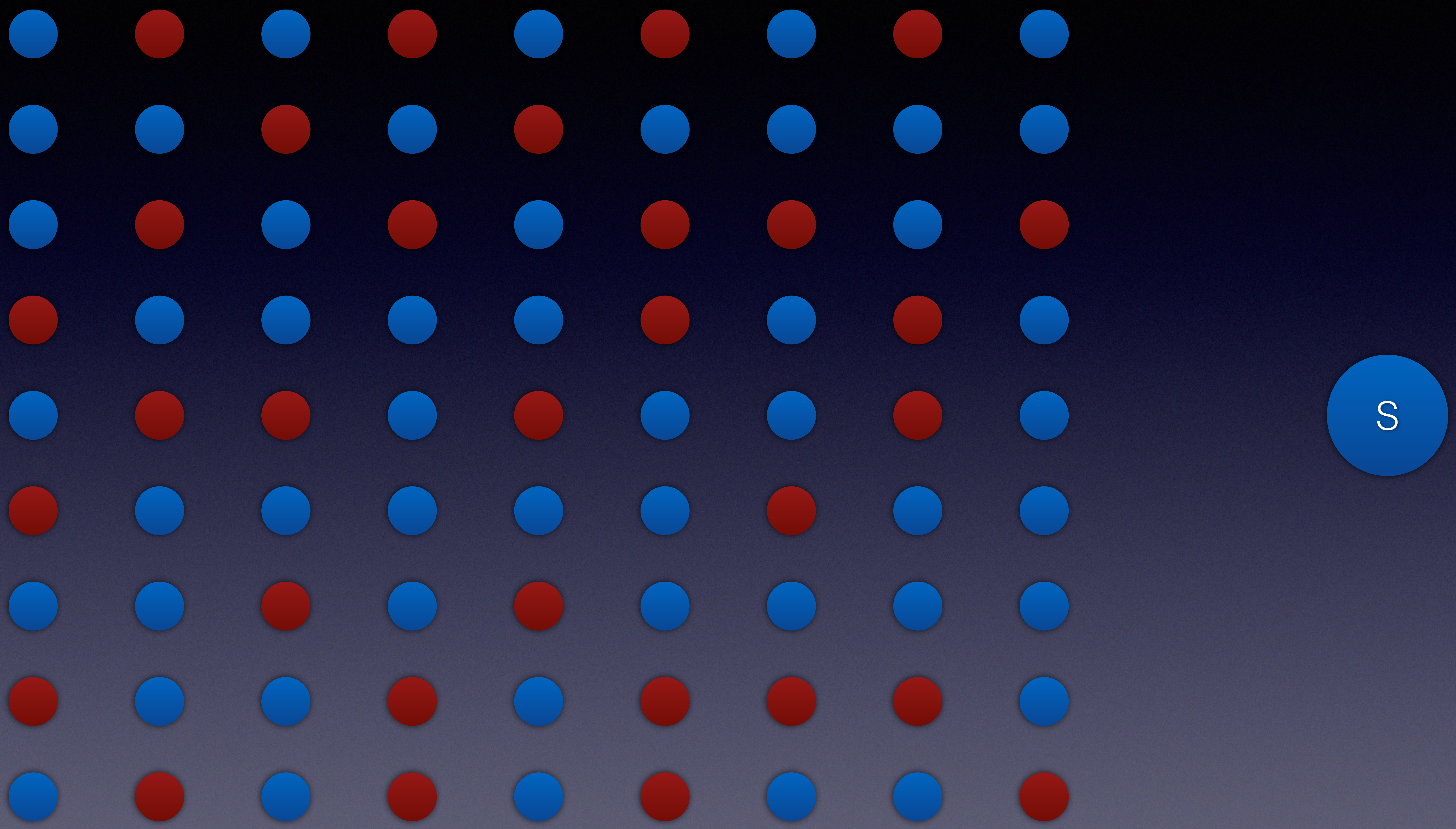
r=a+3



$r=a+4$



$r=a+5$

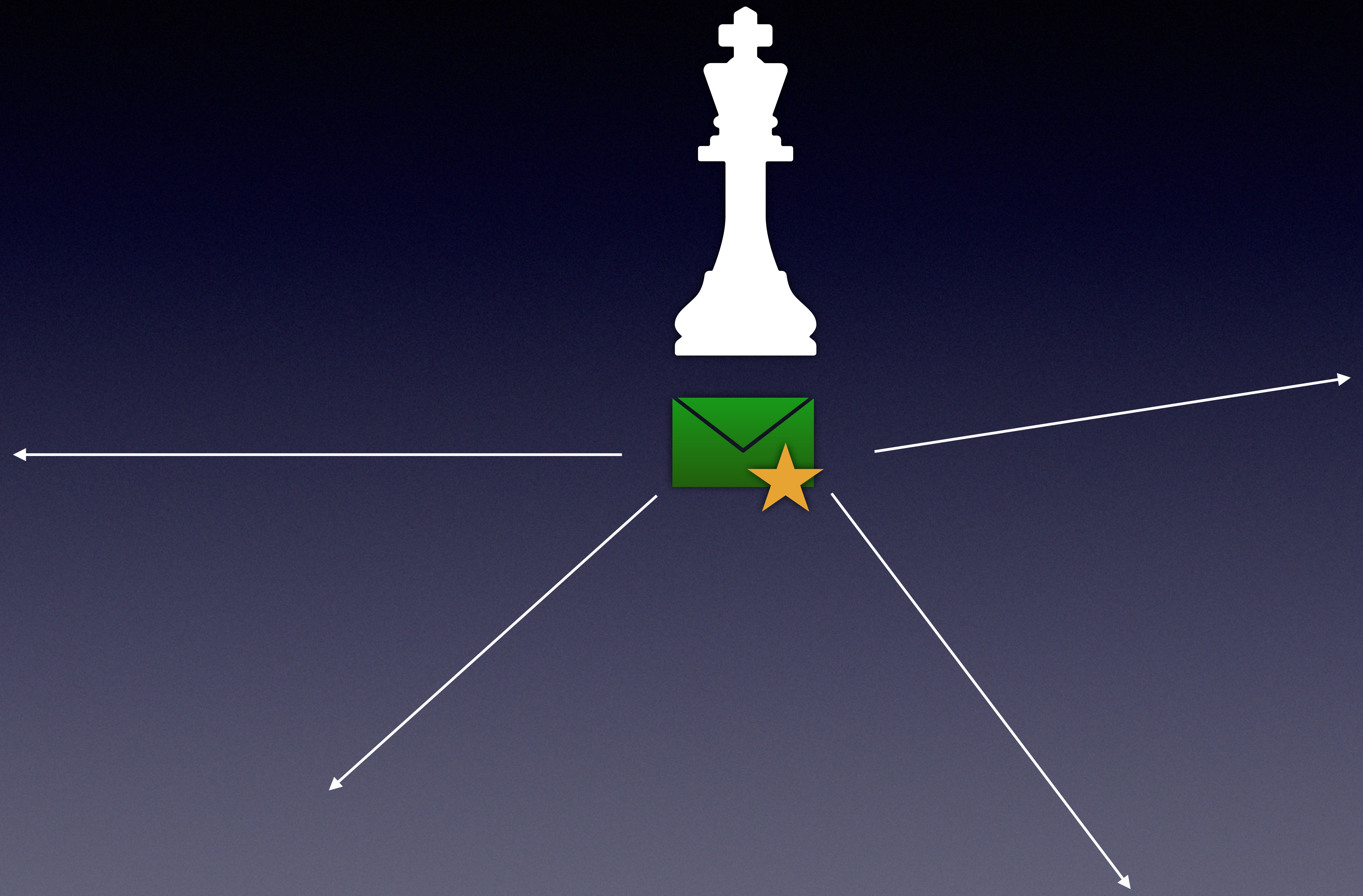


Our Idea for BC

- Gossiping:
  - Each honest party picks randomly ~$\mathcal{O}(\log n)$ other parties to send to.
  - (Ofc, this doesn't work single-shot.) Takes ~$\mathcal{O}(\log n)$ rounds.

BulletinBC Protocol

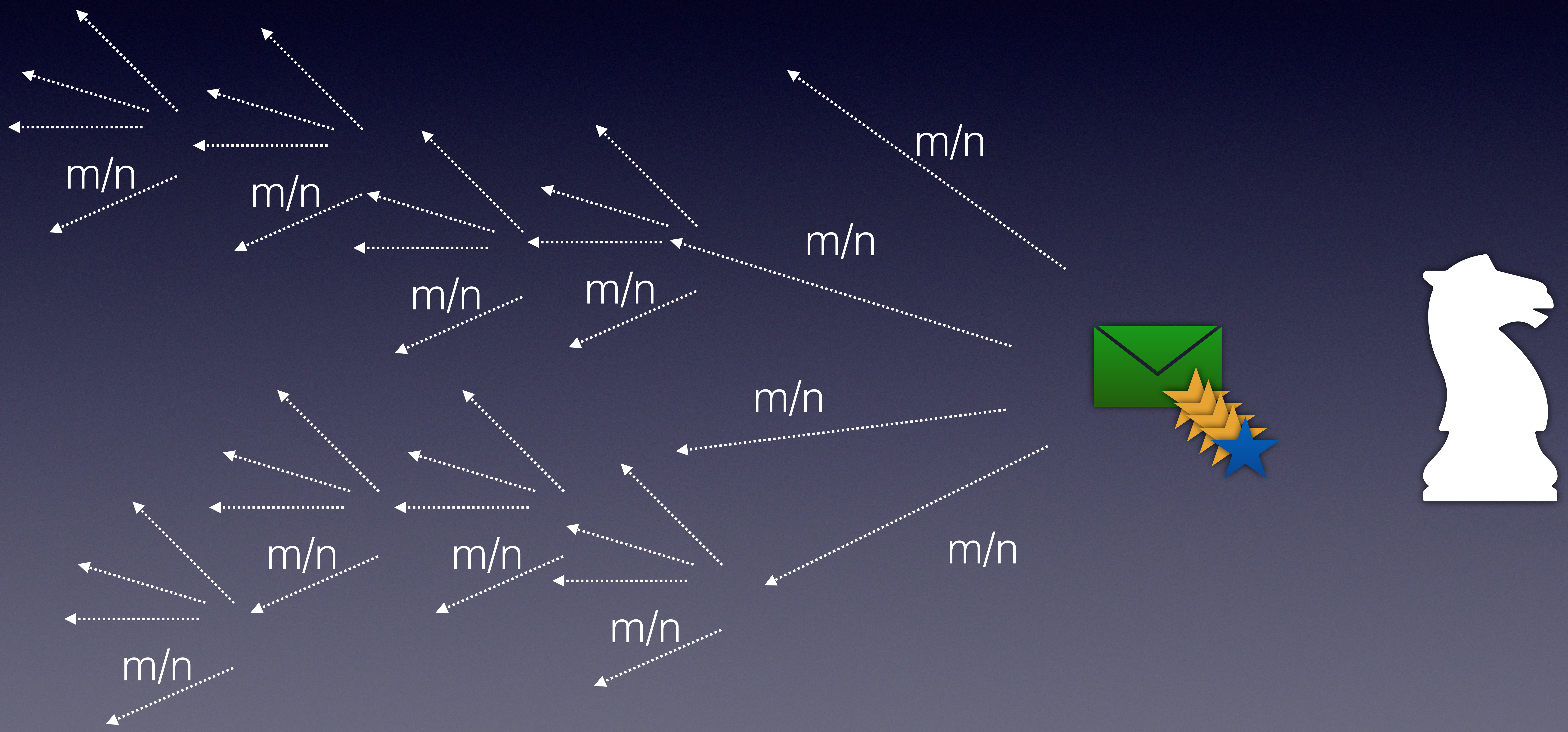
S sends ✉ with S’s signature ★ to all parties



For each $r \leq t+1$:

p checks if it received some new "valid"

If so, it adds its signature and gossips …



Result

Result

- Achieved Authenticated Broadcast:

Result

- Achieved Authenticated Broadcast:
  - For any $t \leq (1 - \epsilon)n$ static corruptions

Result

- Achieved Authenticated Broadcast:
  - For any $t \leq (1-\epsilon)n$ static corruptions
  - Randomized, in $t \cdot \mathcal{O}(\log n) = \mathcal{O}(n \cdot \log n)$ rounds

Result

- Achieved Authenticated Broadcast:
  - For any $t \leq (1-\epsilon)n$ static corruptions
  - Randomized, in $t \cdot \mathcal{O}(\log n) = \mathcal{O}(n \cdot \log n)$ rounds
    - ★ The actual protocol achieves improved $t + \log(n-t) + 1 = \mathcal{O}(n)$ rounds

Result

- Achieved Authenticated Broadcast:
  - For any $t \leq (1-\epsilon)n$ static corruptions
  - Randomized, in $t \cdot \mathcal{O}(\log n) = \mathcal{O}(n \cdot \log n)$ rounds
    - ★ The actual protocol achieves improved $t + \log(n-t) + 1 = \mathcal{O}(n)$ rounds
  - Assumes only bulletin board PKI

Result

- Achieved Authenticated Broadcast:
  - For any $t \leq (1-\epsilon)n$ static corruptions
  - Randomized, in $t \cdot \mathcal{O}(\log n) = \mathcal{O}(n \cdot \log n)$ rounds
    - ★ The actual protocol achieves improved $t + \log(n-t) + 1 = \mathcal{O}(n)$ rounds
  - Assumes only bulletin board PKI
  - With $\tilde{\mathcal{O}}(n^2\kappa^2)$ Communication

Comparison

- Bulletin-Board PKI (**NO trusted setup**)
- State-of-the-art **Communication Complexity** for $t > n/2$

Protocol	Model	CC	RC	Adversary	Corruptions	Type
Dolev-Strong	Bulletin	$O(n^3\kappa)$	$O(n)$	Adaptive	$< n$	BC
BulletinBC	Bulletin	$\tilde{O}(n^2\kappa^2)$	$O(n)$	Static	$< (1-\epsilon)n$	BC
Abraham et al.	Trusted	$\tilde{O}(n\kappa)$	$O(1)$	Adaptive	$< n/2$	BC
Chan et al.	Trusted	$O(n^2\kappa^2)$	$O(\kappa)$	Adaptive	$< (1-\epsilon)n$	BC
Momose and Ren	Bulletin	$\tilde{O}(n^2\kappa)$	$O(n)$	Adaptive	$< n/2$	BC

Comparison

- Bulletin-Board PKI (**NO trusted setup**)
- State-of-the-art **Communication Complexity** for $t > n/2$

Protocol	Model	CC	RC	Adversary	Corruptions	Type
Dolev-Strong	Bulletin	$O(n^3\kappa)$	$O(n)$	Adaptive	$< n$	BC
BulletinBC	Bulletin	$\tilde{O}(n^2\kappa^2)$	$O(n)$	Static	$< (1-\epsilon)n$	BC
Abraham et al.	Trusted	$\tilde{O}(n\kappa)$	$O(1)$	Adaptive	$< n/2$	BC
Chan et al.	Trusted	$O(n^2\kappa^2)$	$O(\kappa)$	Adaptive	$< (1-\epsilon)n$	BC
Momose and Ren	Bulletin	$\tilde{O}(n^2\kappa)$	$O(n)$	Adaptive	$< n/2$	BC

Limitations so far

- **Static** vs **Adaptive** adversary:
- An adaptive adversary can break the security of the process.
  - Any ideas how?

But… Broadcast?

- Back to our motivation:
  - Many times in practical uses of Broadcast, we require **all parties to broadcast** values.
    - (E.g. MPC, VSS applications)

Parallel Broadcast

Parallel Broadcast

- n parties, t corrupted, each party p_i has input bit b_i

Parallel Broadcast

- n parties, t corrupted, each party p_i has input bit b_i
- Each party p_i defines a "slot" s_i

Parallel Broadcast

- n parties, t corrupted, each party p_i has input bit b_i
- Each party p_i defines a "slot" s_i
- Each party p_i outputs a vector of n bits $B_i = (b_1^i, \ldots, b_n^i)$

Parallel Broadcast

- n parties, t corrupted, each party p_i has input bit b_i
- Each party p_i defines a "slot" s_i
- Each party p_i outputs a vector of n bits $B_i = (b_1^i, \ldots, b_n^i)$
- Properties:

Parallel Broadcast

- n parties, t corrupted, each party p_i has input bit b_i
- Each party p_i defines a "slot" s_i
- Each party p_i outputs a vector of n bits $B_i = (b_1^i, \ldots, b_n^i)$
- Properties:
  - Validity: For each "honest slot" s_i all honest parties agree on b_i

Parallel Broadcast

- n parties, t corrupted, each party p_i has input bit b_i
- Each party p_i defines a "slot" s_i
- Each party p_i outputs a vector of n bits $B_i = (b_1^i, \ldots, b_n^i)$
- Properties:
  - Validity: For each "honest slot" s_i all honest parties agree on b_i
  - Consistency: For each slot s_i, all honest parties output the same bit

1. Caesar

5. Alexander

2. Washington

4. Napoleon

3. Charlemagne

1. Caesar

$[b_1, \mathsf{sig}_1(b_1)]$

2. Washington

$[b_2, \mathsf{sig}_2(b_2)]$

3. Charlemagne

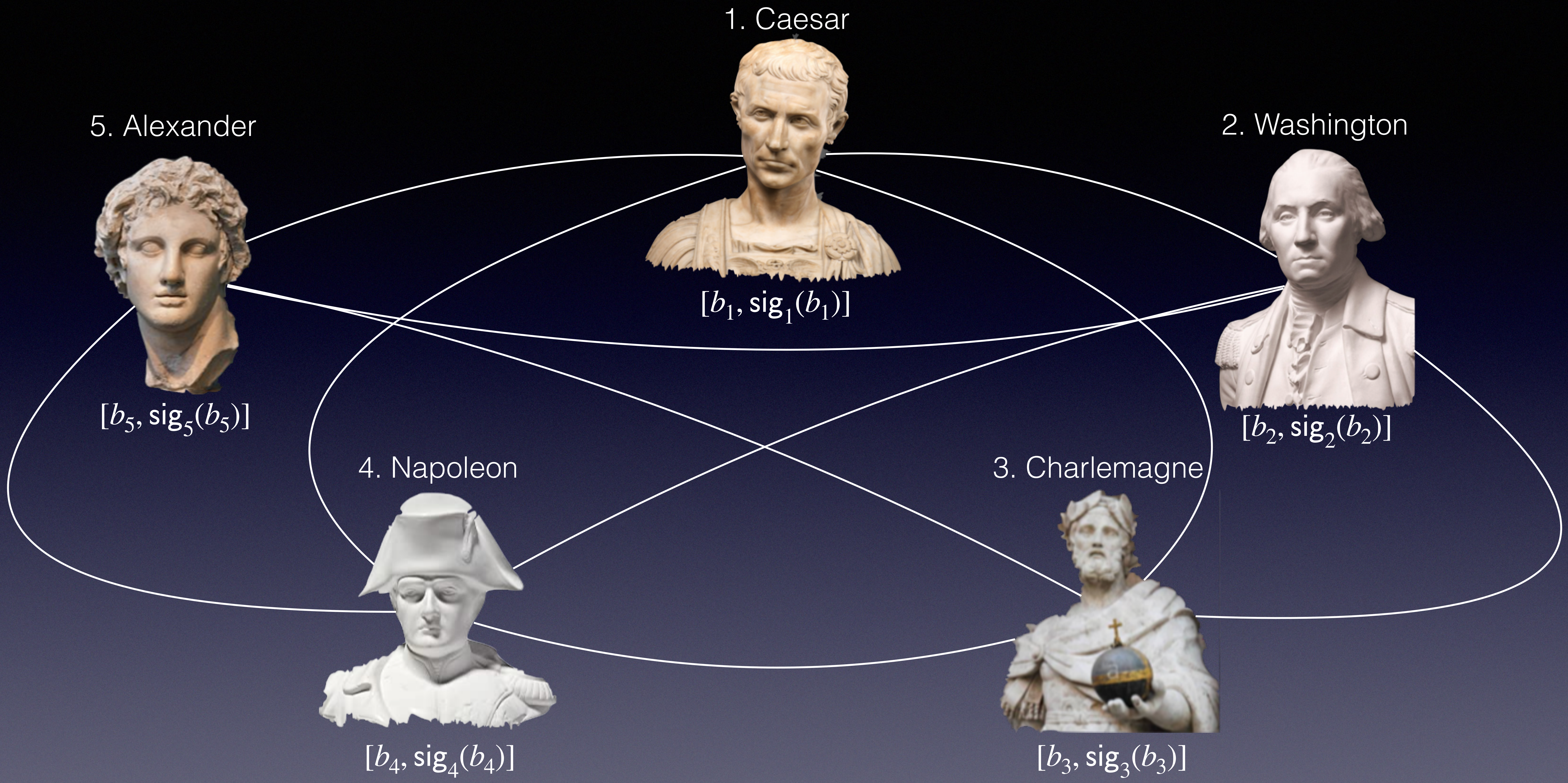
$[b_3, \mathsf{sig}_3(b_3)]$

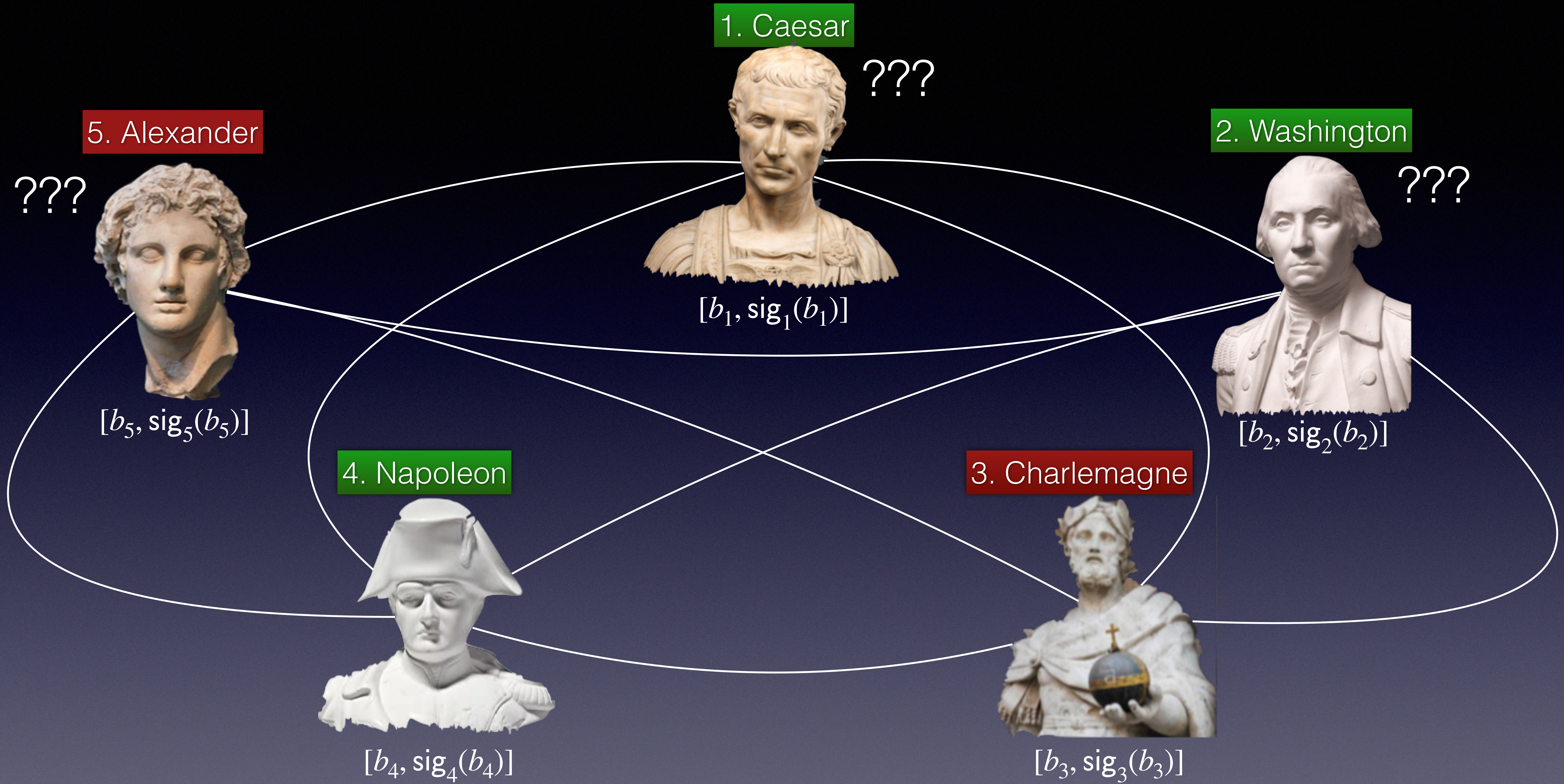
4. Napoleon

$[b_4, \mathsf{sig}_4(b_4)]$

5. Alexander

$[b_5, \mathsf{sig}_5(b_5)]$


1. Caesar
$[b_1, \mathsf{sig}_1(b_1)]$
2. Washington
$[b_2, \mathsf{sig}_2(b_2)]$
3. Charlemagne
$[b_3, \mathsf{sig}_3(b_3)]$
4. Napoleon
$[b_4, \mathsf{sig}_4(b_4)]$
5. Alexander
$[b_5, \mathsf{sig}_5(b_5)]$


1. Caesar
???
$[b_1, \mathsf{sig}_1(b_1)]$
2. Washington
???
$[b_2, \mathsf{sig}_2(b_2)]$
3. Charlemagne
$[b_3, \mathsf{sig}_3(b_3)]$
4. Napoleon
$[b_4, \mathsf{sig}_4(b_4)]$
5. Alexander
???
$[b_5, \mathsf{sig}_5(b_5)]$

Parallel Broadcast

Parallel Broadcast

- Trivial solution: Use the best Broadcast protocol for the underlying assumptions n **times in parallel**.

Parallel Broadcast

- Trivial solution: Use the best Broadcast protocol for the underlying assumptions n **times in parallel**.
    - If C is the Communication of the Broadcast protocol:

Parallel Broadcast

- Trivial solution: Use the best Broadcast protocol for the underlying assumptions n **times in parallel**.
  - If C is the Communication of the Broadcast protocol:
  - Then, overall Communication: $\mathcal{O}(n \cdot C)$

Parallel Broadcast

- Trivial solution: Use the best Broadcast protocol for the underlying assumptions n **times in parallel**.
  - If C is the Communication of the Broadcast protocol:
  - Then, overall Communication: $\mathcal{O}(n \cdot C)$
- Can we do **better**?

~~Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions.~~

~~Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using~~ **~~bulletin board PKI~~**, ~~against $t < (1-\epsilon)n$~~ **~~static~~** ~~corruptions.~~

~~We introduce~~ **~~gossiping~~** and **Converge** and show Parallel Broadcast with:

~~Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions.~~

~~We introduce **gossiping**~~ and **Converge** and show Parallel Broadcast with:

$\mathcal{O}(n^3)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions.

~~Authenticated Broadcast with $\mathcal{O}(n^2)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions.~~

~~We introduce **gossiping**~~ and Converge and show Parallel Broadcast with:

$\mathcal{O}(n^3)$ CC using **bulletin board PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions.

$\mathcal{O}(n^2)$ CC using **trusted PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions.

Converge

Converge

- In PBC, parties have to propagate at least O(n) messages total (1 message per party)

Converge

- In PBC, parties have to propagate at least O(n) messages total (1 message per party)
- We can combine this inherent amount of messages with gossiping to achieve PBC:

Converge

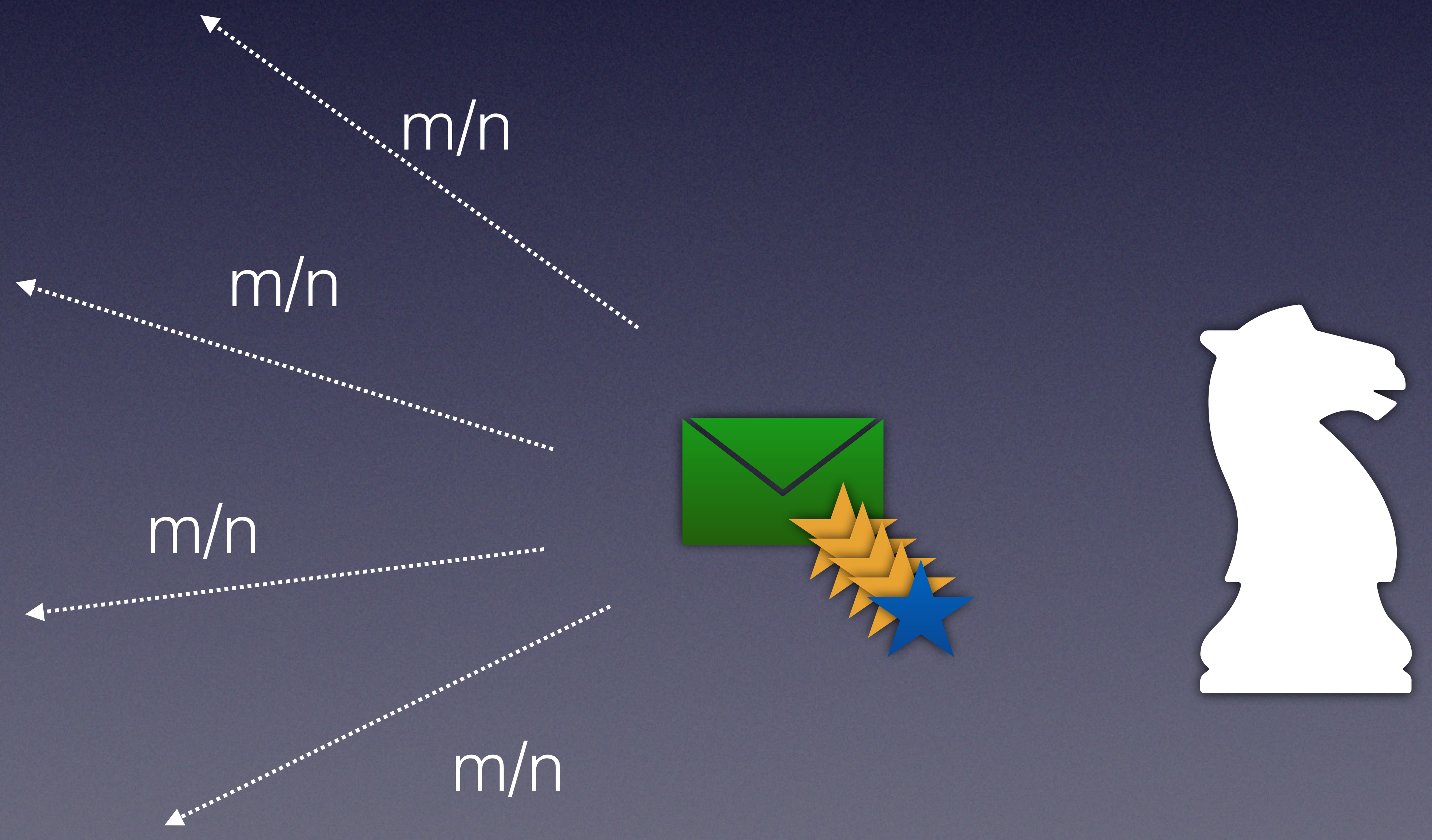
- In PBC, parties have to propagate at least O(n) messages total (1 message per party)
- We can combine this inherent amount of messages with gossiping to achieve PBC:
  - Efficiently

Converge

- In PBC, parties have to propagate at least O(n) messages total (1 message per party)
- We can combine this inherent amount of messages with gossiping to achieve PBC:
  - Efficiently
  - Against adaptive adversaries

Converge

- Before: Party gossips a specific message to a few other parties randomly



Converge

Converge

- Now: Party has to send many ($\Omega(n)$) messages in worst case

Converge

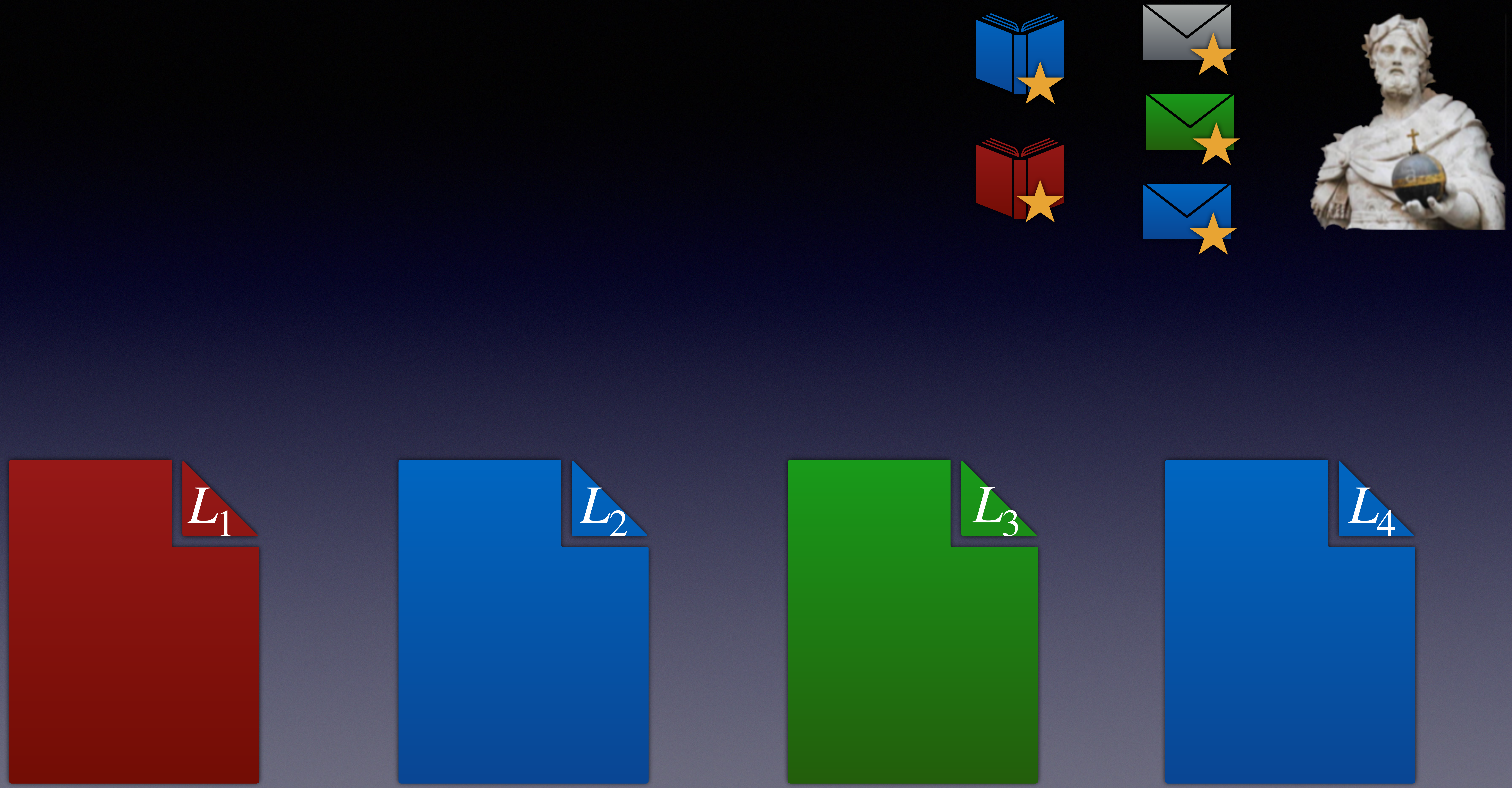
- Now: Party has to send many ($\Omega(n)$) messages in worst case
- Like in gossiping, for each message, randomly select a few parties to send it to

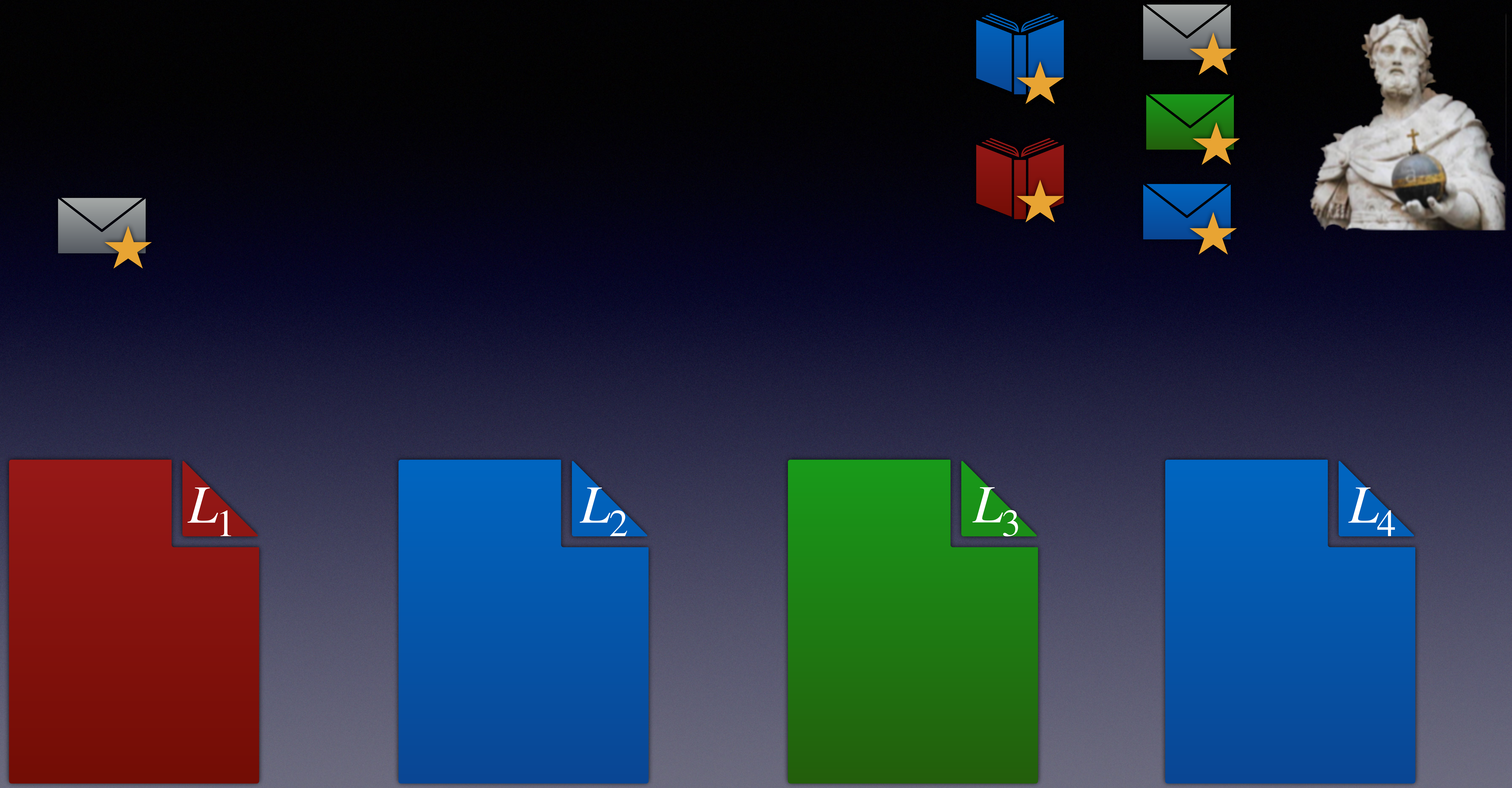
Converge

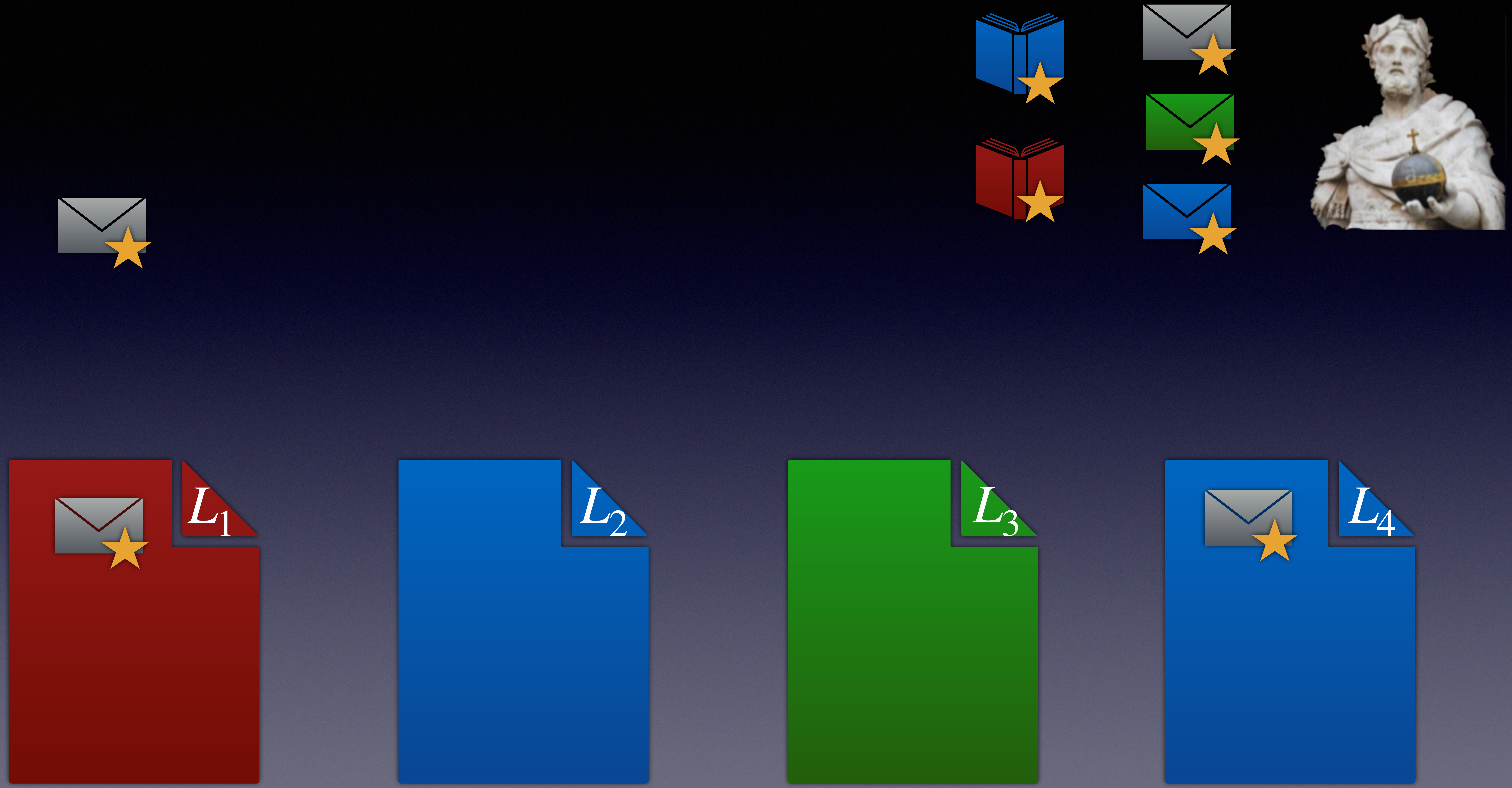
- Now: Party has to send many ($\Omega(n)$) messages in worst case
- Like in gossiping, for each message, randomly select a few parties to send it to
- With high prob. all parties receive ~ the same amount of messages

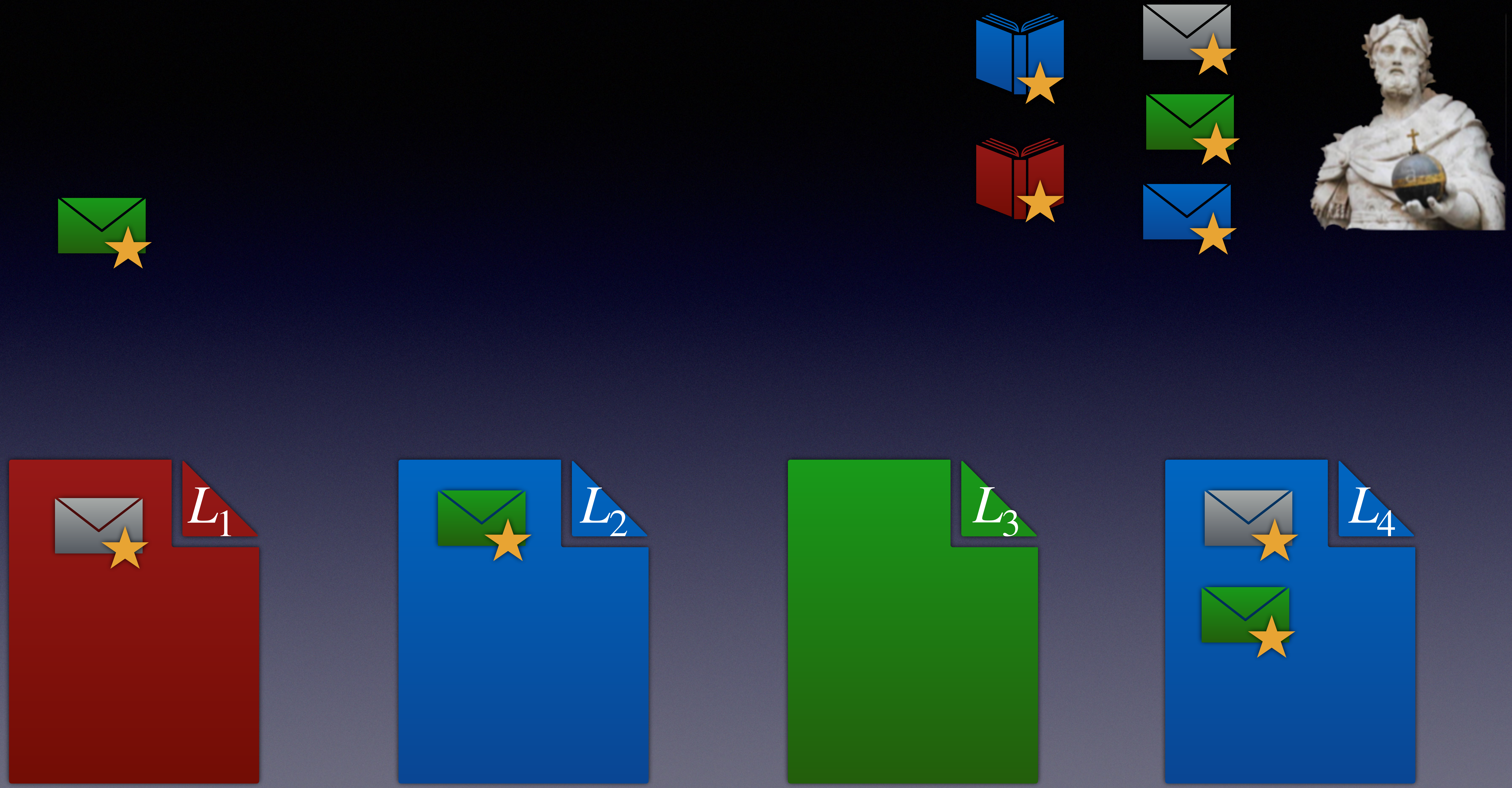
Converge

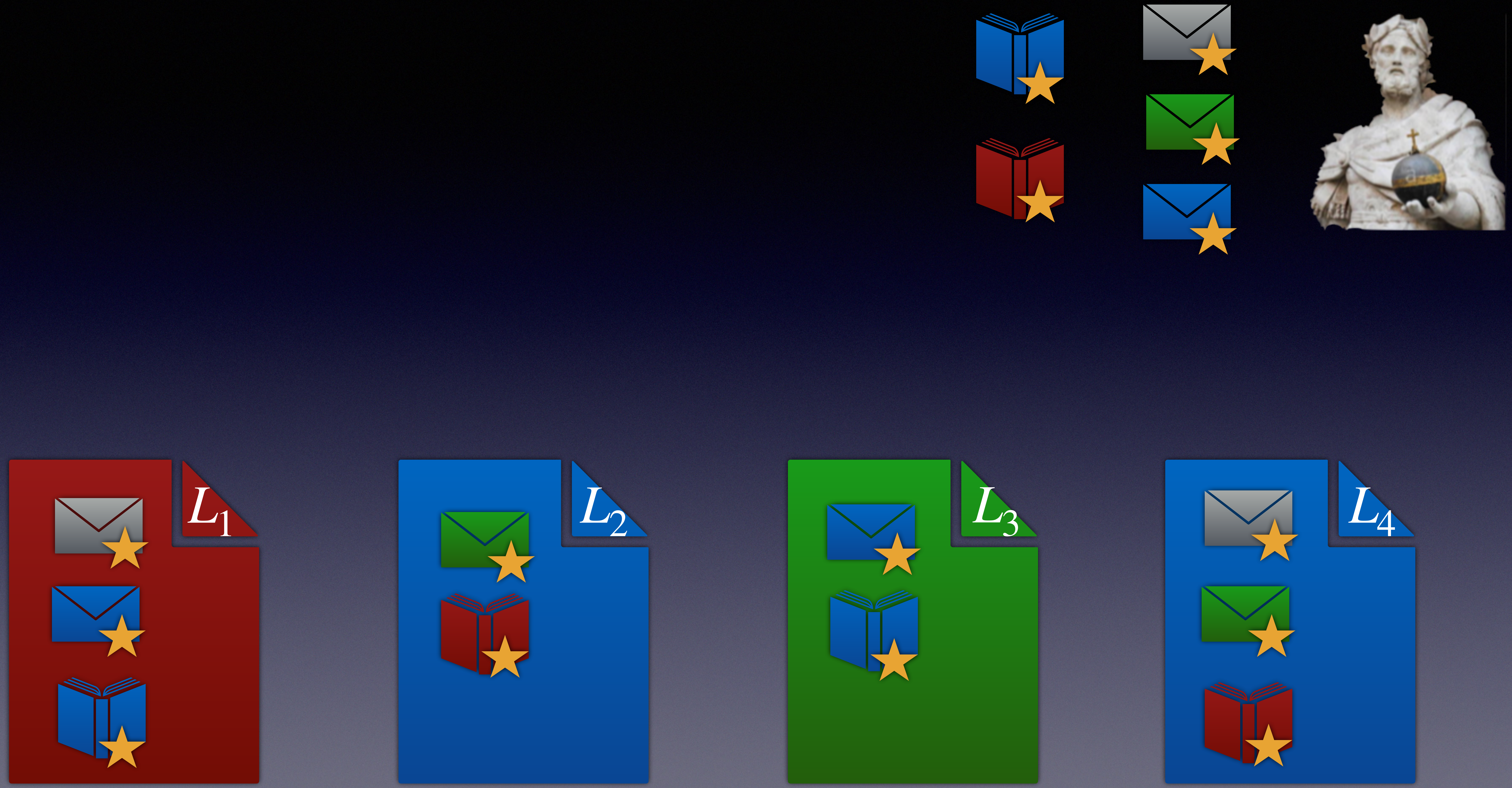
- Now: Party has to send many ($\Omega(n)$) messages in worst case
- Like in gossiping, for each message, randomly select a few parties to send it to
- With high prob. all parties receive ~ the same amount of messages
- The adversary doesn't gain any advantage by observing the execution of the protocol

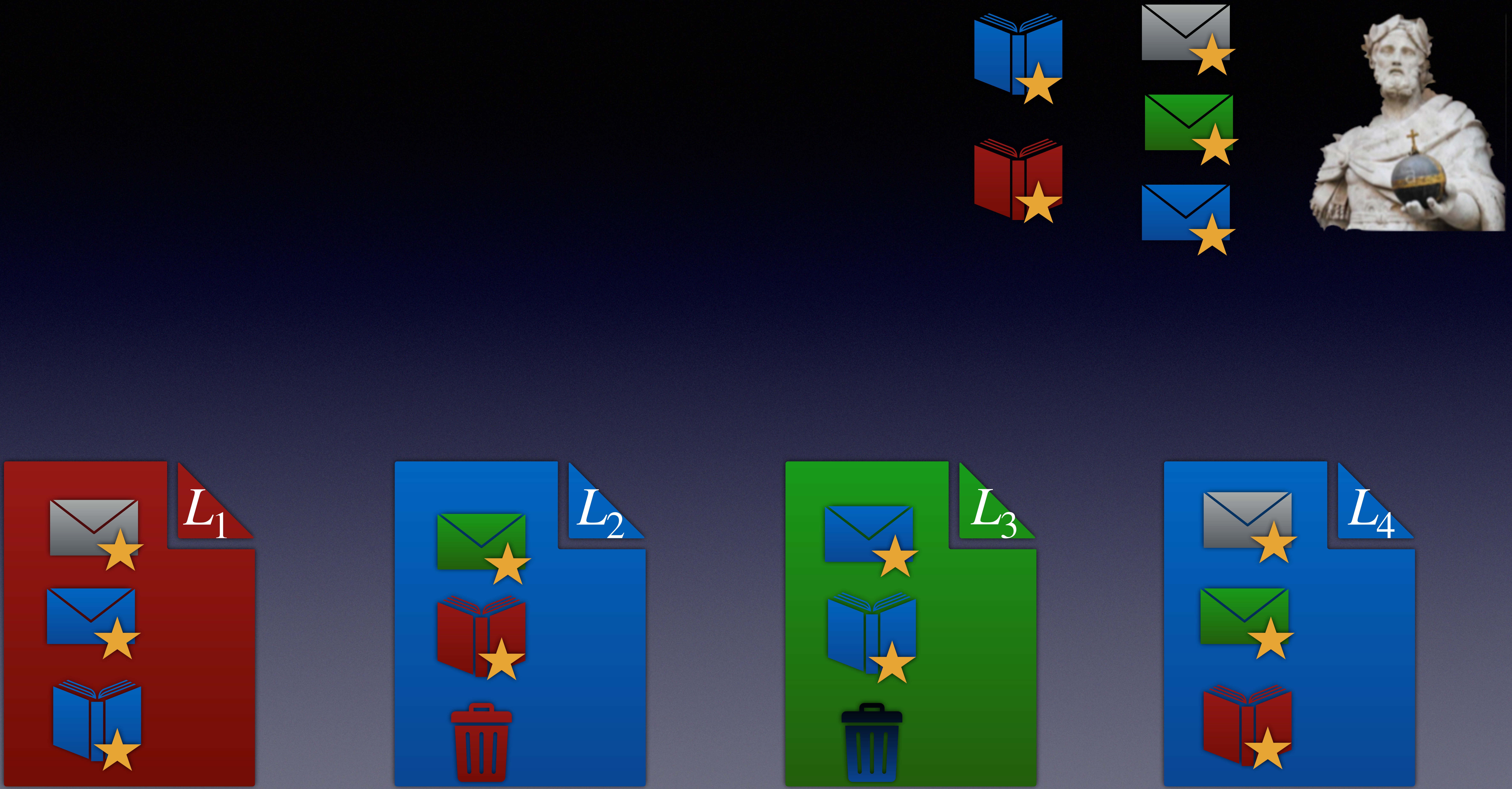

L_1
L_2
L_3
L_4

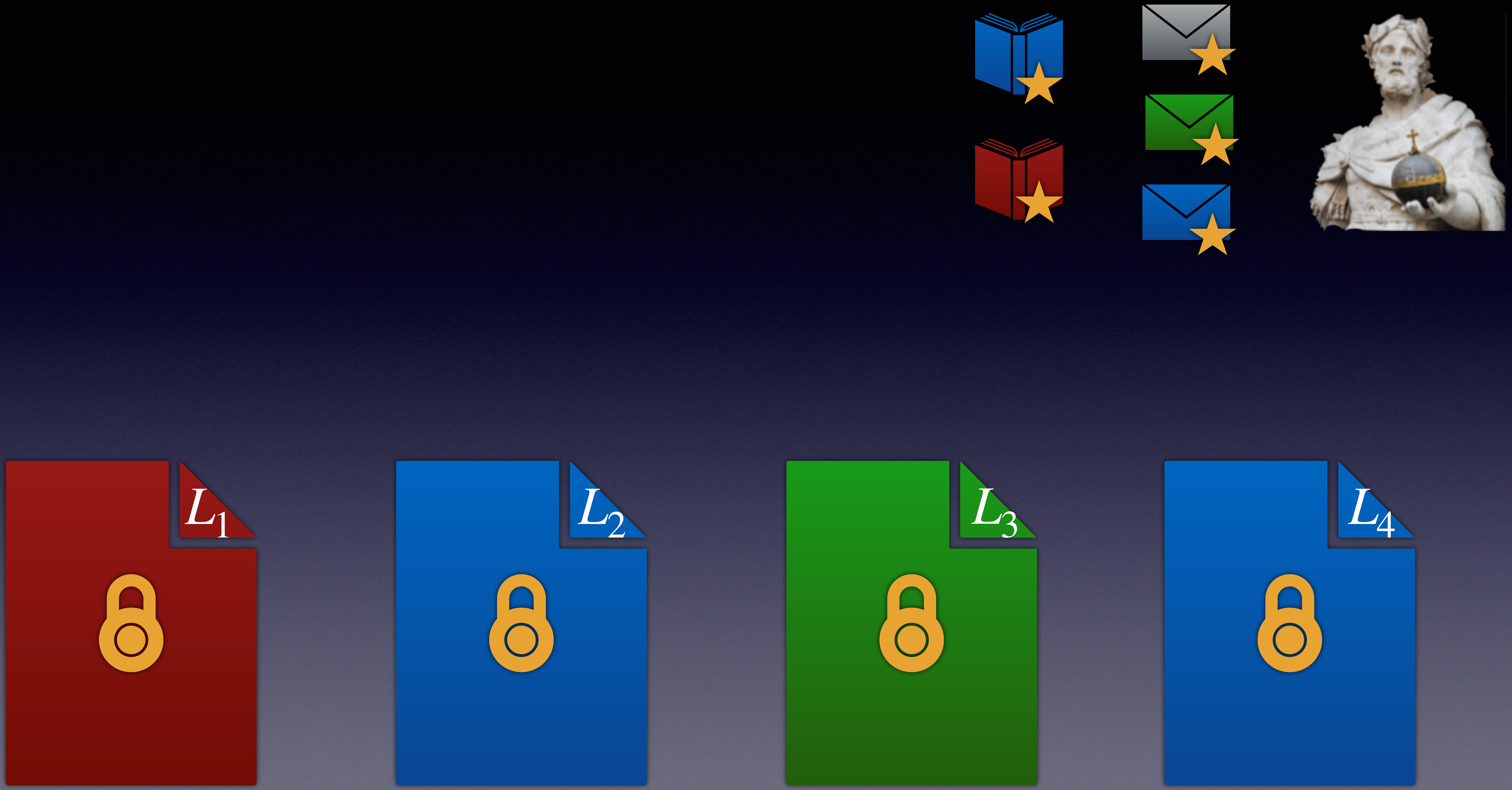

L_1
L_2
L_3
L_4


L_1
L_2
L_3
L_4


L_1
L_2
L_3
L_4

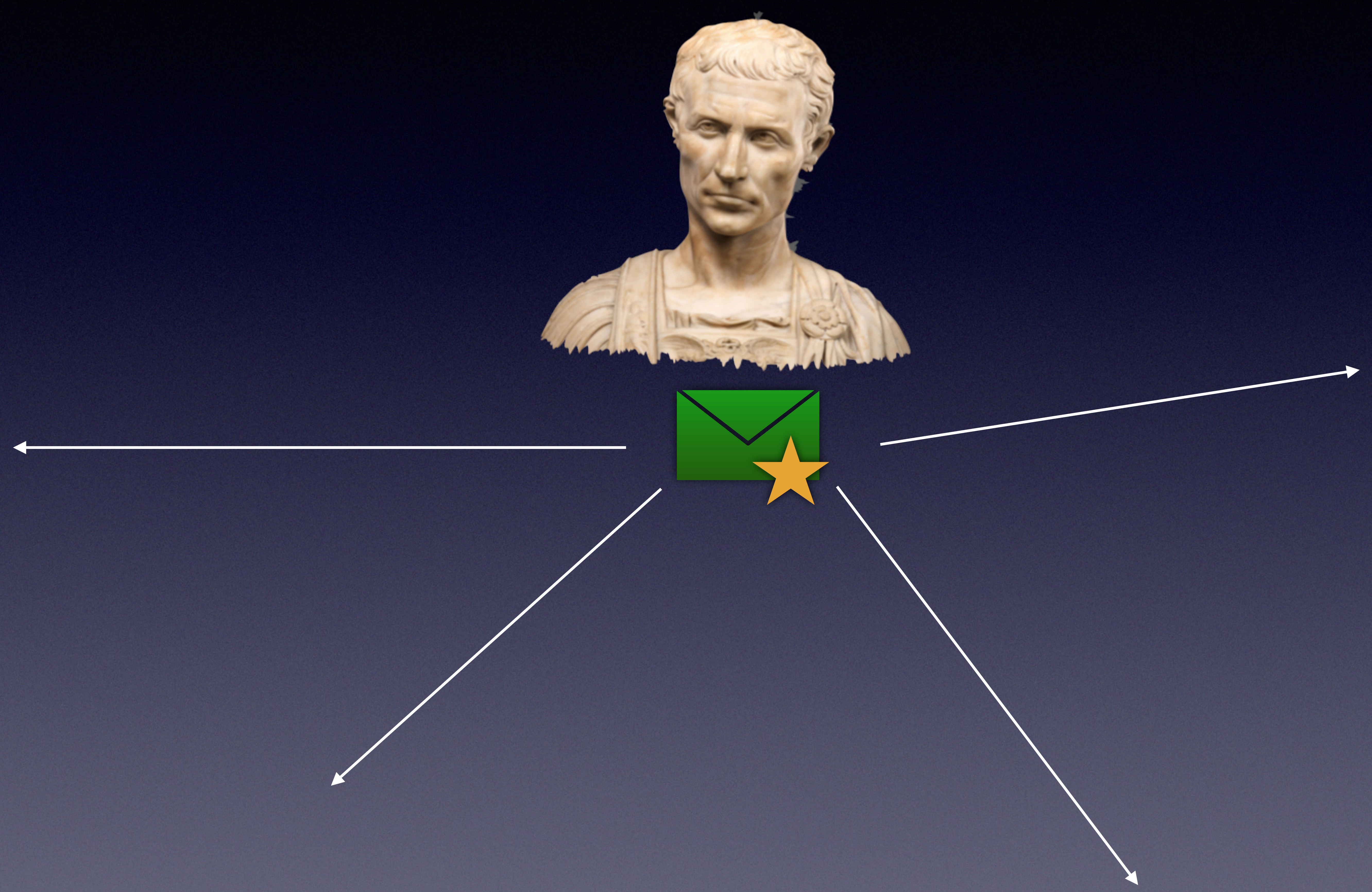

L_1
L_2
L_3
L_4


L_1
L_2
L_3
L_4


L_1
L_2
L_3
L_4

Our Bulletin PBC Protocol

Each party S sends with S’s signature to all parties



Stage 1:

For each $r \leq t + 1$:
p checks if it received some new "**valid**" bit.
For such bit b and slot s, **add** $sig_p([b, s])$ ★.

Stage 1:

For each $r \leq t + 1$:
p checks if it received some new “**valid**” bit.
For such bit b and slot s, **add** $sig_p([b, s])$ ★.

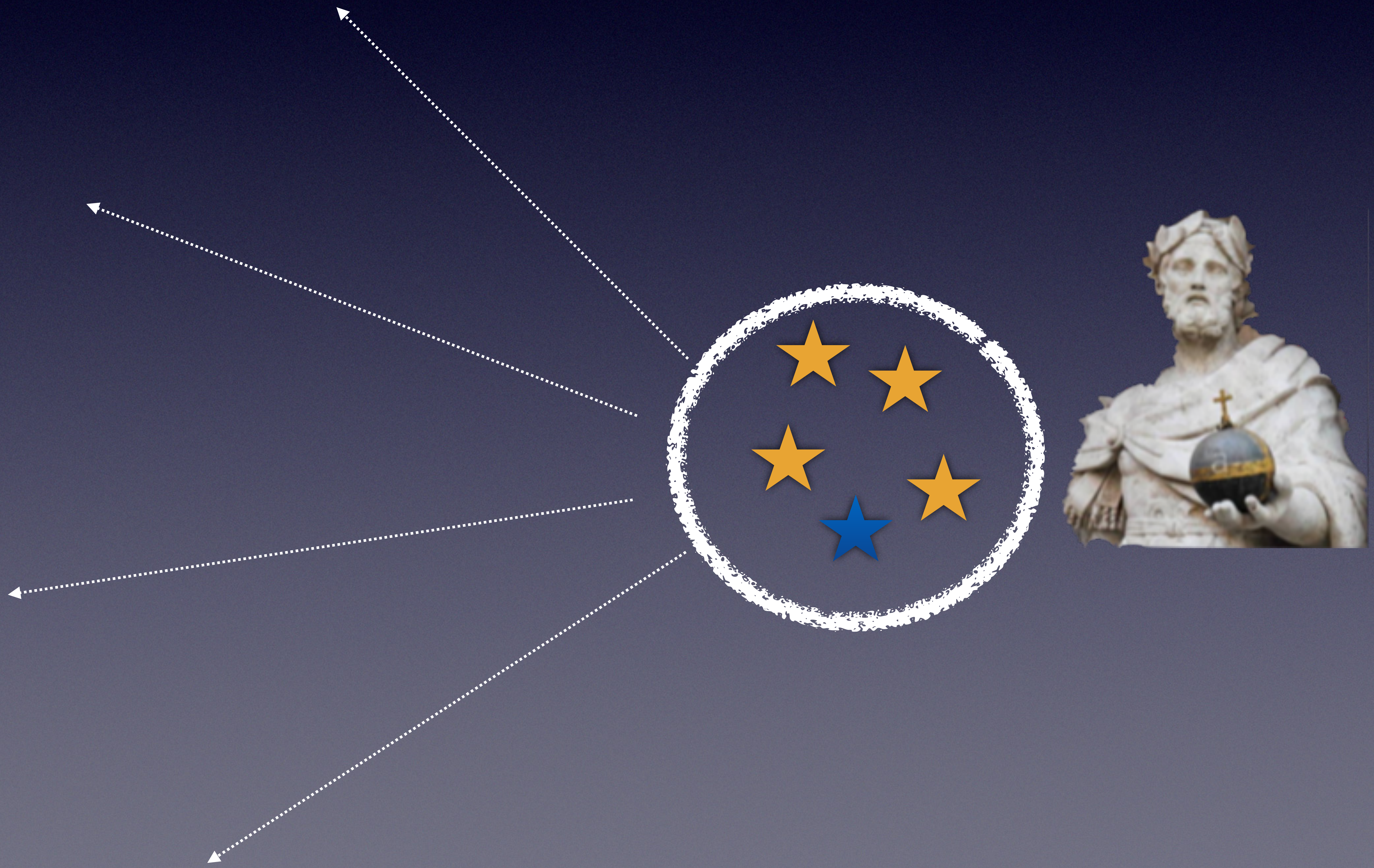
Valid bit at round r:

At least r distinct signatures, where one is from s.

Stage 2:

For each $r \leq t + 1$:

p calls **Converge** on M_p : received signatures



PBC without trusted setup

PBC without trusted setup

- Communication: O(n) rounds, each round calls **Converge**

PBC without trusted setup

- Communication: O(n) rounds, each round calls **Converge**
- Message space $\mathscr{M}$: signatures on [b,s], $|\mathscr{M}| = O(n^2)$

PBC without trusted setup

- Communication: O(n) rounds, each round calls **Converge**
- Message space $\mathscr{M}$: signatures on [b,s], $|\mathscr{M}| = O(n^2)$
- Optimization: p propagates each signature in $O(1)$ rounds

PBC without trusted setup

- Communication: O(n) rounds, each round calls **Converge**
- Message space $\mathscr{M}$: signatures on [b,s], $|\mathscr{M}| = O(n^2)$
- Optimization: p propagates each signature in $O(1)$ rounds
- Total CC: $\tilde{O}(n^3)$ (Amortized $\tilde{O}(n^2)$ per broadcast)

Our bulletin board PBC Result

Our bulletin board PBC Result

- Achieved Authenticated Parallel Broadcast:

Our bulletin board PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1 - \epsilon)n$ adaptive corruptions

Our bulletin board PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1 - \epsilon)n$ adaptive corruptions
  - Randomized, $\mathcal{O}(n \log n)$ rounds

Our bulletin board PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1-\epsilon)n$ adaptive corruptions
  - Randomized, $\mathcal{O}(n \log n)$ rounds
  - Only bulletin board PKI

Our bulletin board PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1-\epsilon)n$ adaptive corruptions
  - Randomized, $\mathcal{O}(n \log n)$ rounds
  - Only bulletin board PKI
  - With $\tilde{\mathcal{O}}(n^3\kappa^2)$ Communication

Our trusted PBC Result

Our trusted PBC Result

- Modified a single-sender Broadcast protocol by Chan et al.[PKC'20]

Our trusted PBC Result

- Modified a single-sender Broadcast protocol by Chan et al.[PKC'20]
- Committee-based

Our trusted PBC Result

- Modified a single-sender Broadcast protocol by Chan et al.[PKC’20]
- Committee-based
- In each round, message propagation follows Converge instead of Send-to-all

Our trusted PBC Result

Our trusted PBC Result

- Achieved Authenticated Parallel Broadcast:

Our trusted PBC Result

- Achieved Authenticated Parallel Broadcast:
    - For any $t \leq (1-\epsilon)n$ adaptive corruptions

Our trusted PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1 - \epsilon)n$ adaptive corruptions
  - Randomized, $\mathcal{O}(\kappa \log n)$ rounds

Our trusted PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1 - \epsilon)n$ adaptive corruptions
  - Randomized, $\mathcal{O}(\kappa \log n)$ rounds
  - Assumes trusted PKI

Our trusted PBC Result

- Achieved Authenticated Parallel Broadcast:
  - For any $t \leq (1-\epsilon)n$ adaptive corruptions
  - Randomized, $\mathcal{O}(\kappa \log n)$ rounds
  - Assumes trusted PKI
  - With $\tilde{\mathcal{O}}(n^2\kappa^4)$ Communication

Comparison

Protocol	Model	CC	RC	Adversary	Corruptions	Type
Dolev-Strong	Bulletin	$O(n^3\kappa)$	$O(n)$	Adaptive	$< n$	BC
BulletinBC	Bulletin	$\tilde{O}(n^2\kappa^2)$	$O(n)$	Static	$< (1-\epsilon)n$	BC
Abraham et al.	Trusted	$\tilde{O}(n\kappa)$	$O(1)$	Adaptive	$< n/2$	BC
Chan et al.	Trusted	$O(n^2\kappa^2)$	$O(\kappa)$	Adaptive	$< (1-\epsilon)n$	BC
Momose and Ren	Bulletin	$\tilde{O}(n^2\kappa)$	$O(n)$	Adaptive	$< n/2$	BC
BulletinPBC	Bulletin	$\tilde{O}(n^2\kappa^2)*$	$O(n \log n)$	Adaptive	$< (1-\epsilon)n$	PBC
TrustedPBC	Trusted	$\tilde{O}(n\kappa^4)*$	$O(\kappa \log n)$	Adaptive	$< (1-\epsilon)n$	PBC

* refers to amortized Complexity per sender

Comparison

Protocol	Model	CC	RC	Adversary	Corruptions	Type
Dolev-Strong	Bulletin	$O(n^3\kappa)$	$O(n)$	Adaptive	$< n$	BC
BulletinBC	Bulletin	$\tilde{O}(n^2\kappa^2)$	$O(n)$	Static	$< (1-\epsilon)n$	BC
Abraham et al.	Trusted	$\tilde{O}(n\kappa)$	$O(1)$	Adaptive	$< n/2$	BC
Chan et al.	Trusted	$O(n^2\kappa^2)$	$O(\kappa)$	Adaptive	$< (1-\epsilon)n$	BC
Momose and Ren	Bulletin	$\tilde{O}(n^2\kappa)$	$O(n)$	Adaptive	$< n/2$	BC
BulletinPBC	Bulletin	$\tilde{O}(n^2\kappa^2)*$	$O(n \log n)$	Adaptive	$< (1-\epsilon)n$	PBC
TrustedPBC	Trusted	$\tilde{O}(n\kappa^4)*$	$O(\kappa \log n)$	Adaptive	$< (1-\epsilon)n$	PBC

$*$ refers to amortized Complexity per sender

Comparison

Protocol	Model	CC	RC	Adversary	Corruptions	Type
Dolev-Strong	Bulletin	$O(n^3\kappa)$	$O(n)$	Adaptive	$< n$	BC
BulletinBC	Bulletin	$\tilde{O}(n^2\kappa^2)$	$O(n)$	Static	$< (1-\epsilon)n$	BC
Abraham et al.	Trusted	$\tilde{O}(n\kappa)$	$O(1)$	Adaptive	$< n/2$	BC
Chan et al.	Trusted	$O(n^2\kappa^2)$	$O(\kappa)$	Adaptive	$< (1-\epsilon)n$	BC
Momose and Ren	Bulletin	$\tilde{O}(n^2\kappa)$	$O(n)$	Adaptive	$< n/2$	BC
BulletinPBC	Bulletin	$\tilde{O}(n^2\kappa^2)*$	$O(n\log n)$	Adaptive	$< (1-\epsilon)n$	PBC
TrustedPBC	Trusted	$\tilde{O}(n\kappa^4)*$	$O(\kappa\log n)$	Adaptive	$< (1-\epsilon)n$	PBC

* refers to amortized Complexity per sender

Contributions

Contributions

Introduced **gossiping**, **Converge** & **3 SOA** protocols:

Contributions

Introduced **gossiping**, **Converge** & **3 SOA** protocols:

$\tilde{\mathcal{O}}(n^2\kappa^4)$ **Parallel Broadcast** using **trusted PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

Contributions

Introduced **gossiping**, **Converge** & **3 SOA** protocols:

$\tilde{\mathcal{O}}(n^2\kappa^4)$ **Parallel Broadcast** using **trusted PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

$\tilde{\mathcal{O}}(n^3\kappa^2)$ **Parallel Broadcast** using **bulletin board PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

Contributions

Introduced **gossiping**, **Converge** & **3 SOA** protocols:

$\tilde{\mathcal{O}}(n^2\kappa^4)$ **Parallel Broadcast** using **trusted PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

$\tilde{\mathcal{O}}(n^3\kappa^2)$ **Parallel Broadcast** using **bulletin board PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

$\tilde{\mathcal{O}}(n^2\kappa^2)$ **single sender Broadcast** using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions

Contributions

Introduced **gossiping**, **Converge** & **3 SOA** protocols:

$\tilde{\mathcal{O}}(n^2\kappa^4)$ **Parallel Broadcast** using **trusted PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

$\tilde{\mathcal{O}}(n^3\kappa^2)$ **Parallel Broadcast** using **bulletin board PKI**, against $t < (1-\epsilon)n$ **adaptive** corruptions

$\tilde{\mathcal{O}}(n^2\kappa^2)$ **single sender Broadcast** using **bulletin board PKI**, against $t < (1-\epsilon)n$ **static** corruptions

Interested in our paper? eprint.iacr.org/2020/894