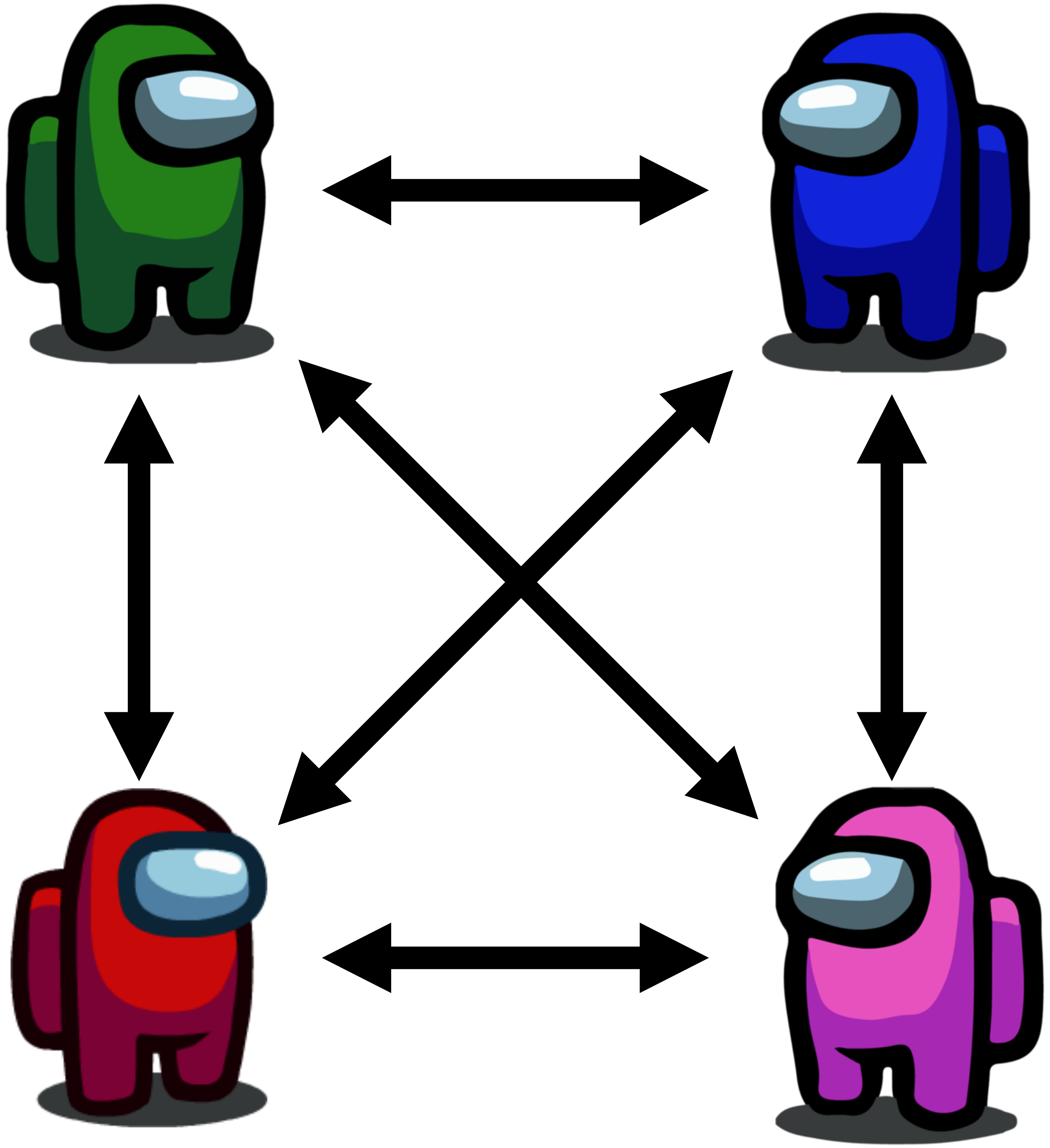
Adaptive Security, Erasures, and Network Assumptions in Communication-Local MPC

Ankit Kumar Misra
UCLA

Joint work with: Nishanth Chandran, Juan Garay, Rafail Ostrovsky, and Vassilis Zikas

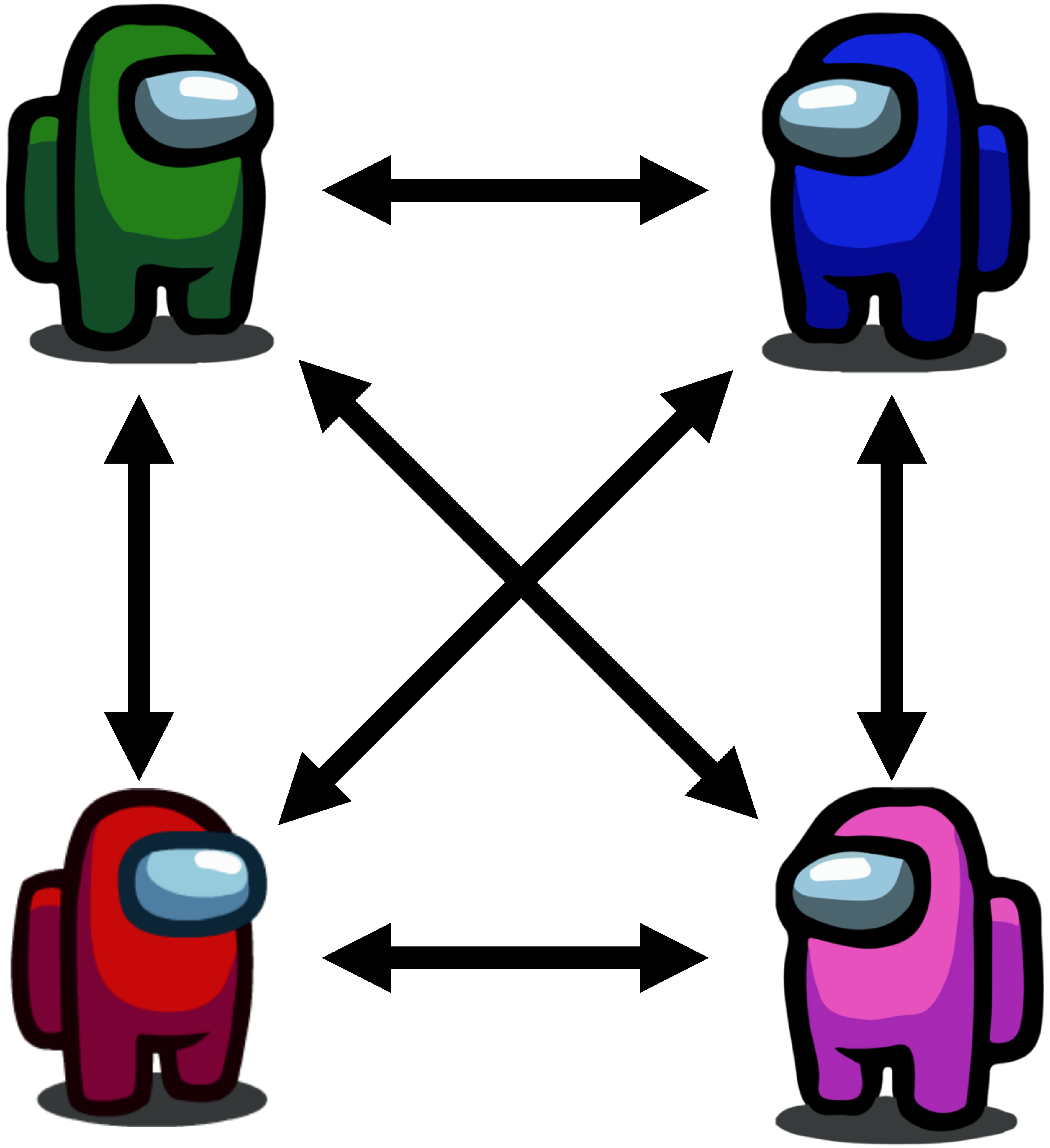
TCC 2024

Complexity Measures in MPC



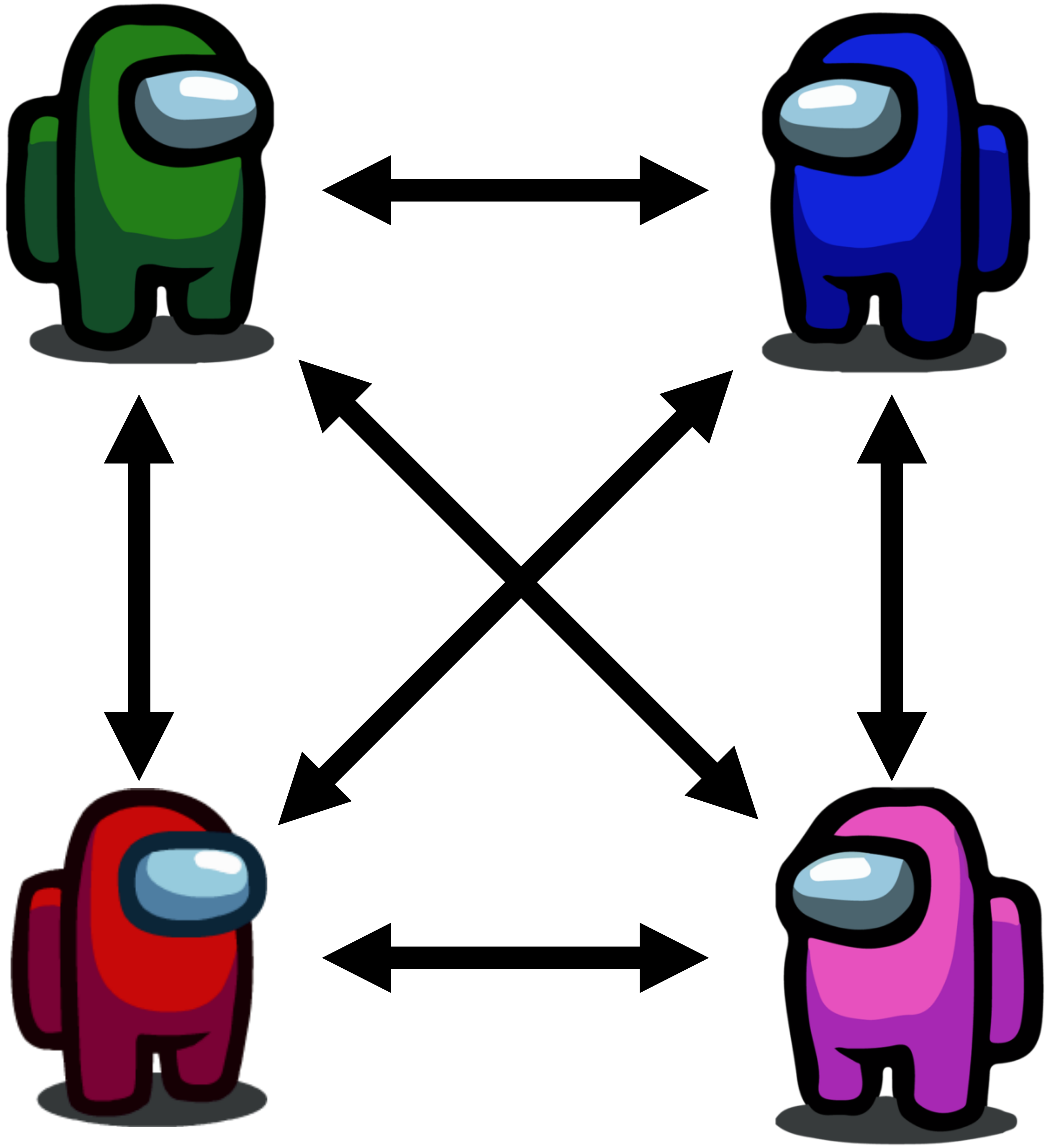
Complexity Measures in MPC

- Commonly studied metrics of efficiency and scalability in MPC:
  - Communication complexity
  - Computational complexity
  - Round complexity



Complexity Measures in MPC

- Commonly studied metrics of efficiency and scalability in MPC:
  - Communication complexity
  - Computational complexity
  - Round complexity
- In standard MPC protocols, every party talks to every other party



Communication Locality

Communication Locality

- Communication Locality [BGT13]: Number of point-to-point channels used by each party

Communication Locality

- Communication Locality [BGT13]: Number of point-to-point channels used by each party
  - Implicit in previous works on "almost-everywhere secure" protocols [DPPU86,Upf92,KSSV06,GO08,CGO10,CGO12]

Communication Locality

- Communication Locality [BGT13]: Number of point-to-point channels used by each party
  - Implicit in previous works on "almost-everywhere secure" protocols [DPPU86,Upf92,KSSV06,GO08,CGO10,CGO12]
- Standard MPC protocols: communication locality = n-1

Communication Locality

- Communication Locality [BGT13]: Number of point-to-point channels used by each party
  - Implicit in previous works on "almost-everywhere secure" protocols [DPPU86,Upf92,KSSV06,GO08,CGO10,CGO12]
- Standard MPC protocols: communication locality = n-1

Question [BGT13]: MPC with low (sublinear in n) communication locality?

Communication Locality

- Communication Locality [BGT13]: Number of point-to-point channels used by each party
  - Implicit in previous works on "almost-everywhere secure" protocols [DPPU86,Upf92,KSSV06,GO08,CGO10,CGO12]
- Standard MPC protocols: communication locality = n-1

polylog(n)

Question [BGT13]: MPC with low (sublinear in n) communication locality?

Communication Locality

- Communication Locality [BGT13]: Number of point-to-point channels used by each party
  - Implicit in previous works on "almost-everywhere secure" protocols [DPPU86,Upf92,KSSV06,GO08,CGO10,CGO12]
- Standard MPC protocols: communication locality = n-1

polylog(n)

Question [BGT13]: MPC with low (sublinear in n) communication locality?

Communication-local MPC (CL MPC)

The Model

The Model

- Complete graph:
  - All n parties are connected by point-to-point channels
  - Dynamically select which channels to use

The Model

- Complete graph:
  - All n parties are connected by point-to-point channels
  - Dynamically select which channels to use

- Synchronous communication

Communication-Local MPC

	Type of Adversary	Assumptions (on operations)			Result: CL MPC is...
		Atomic Multisend-and-Erase	Erasures	Atomic Multisend	
[BGT13]	Static	×	×	×	**feasible** for $t < (1/3 - \varepsilon)n$ corruptions
[CCG+15]	Adaptive	✓	✓	✓	**feasible** for $t < n/2$ corruptions
This work	Adaptive	×	×	✓	**impossible** for linear corruption, using "store-and-forward" protocols
	Adaptive	×	✓	×	**feasible** for $t < (1/2 - \varepsilon)n$ corruptions (weak cryptographic assumptions)
	Adaptive	×	✓	×	**impossible** for $t > (1/2 + \varepsilon)n$ corruptions, using "store-and-forward" protocols
	Adaptive	×	×	×	**feasible** for $t < n/2$ corruptions (strong cryptographic assumptions)

Communication-Local MPC

	Type of Adversary	Assumptions (on operations)			Result: CL MPC is...
		Atomic Multisend-and-Erase	Erasures	Atomic Multisend	
[BGT13]	Static	×	×	×	**feasible** for $t < (1/3 - \varepsilon)n$ corruptions
[CCG+15]	Adaptive	✓	✓	✓	**feasible** for $t < n/2$ corruptions
This work	Adaptive	×	×	✓	**impossible** for linear corruption, using "store-and-forward" protocols
	Adaptive	×	✓	×	**feasible** for $t < (1/2 - \varepsilon)n$ corruptions (weak cryptographic assumptions)
	Adaptive	×	✓	×	**impossible** for $t > (1/2 + \varepsilon)n$ corruptions, using "store-and-forward" protocols
	Adaptive	×	×	×	**feasible** for $t < n/2$ corruptions (strong cryptographic assumptions)

Background: Ideas from [CCG+15]

Background: Ideas from [CCG+15]

Hidden graph

Background: Ideas from [CCG+15]

Hidden graph

- Adaptive security is impossible if adversary knows communication graph

Background: Ideas from [CCG+15]

Hidden graph

- Adaptive security is impossible if adversary knows communication graph
- [CCG+15]: “hidden graph” setup!

Background: Ideas from [CCG+15]

Hidden graph

- Adaptive security is impossible if adversary knows communication graph
- [CCG+15]: "hidden graph" setup!
  - Each party sees only its neighborhood

Background: Ideas from [CCG+15]

Hidden graph

- Adaptive security is impossible if adversary knows communication graph
- [CCG+15]: "hidden graph" setup!
  - Each party sees only its neighborhood
  - Setup: Use a symmetric key infrastructure — every pair of parties decides if they have edge

Background: Ideas from [CCG+15]

Hidden graph

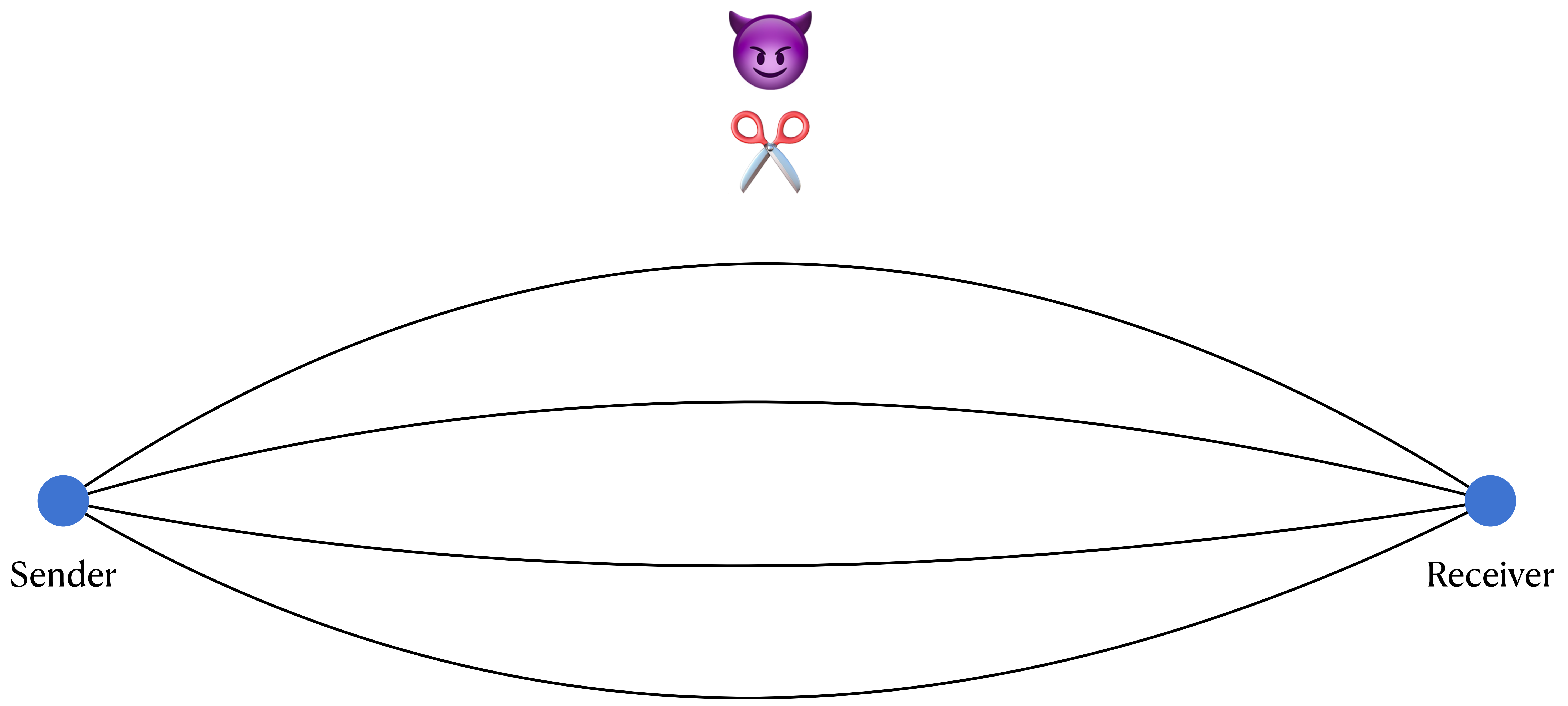
- Adaptive security is impossible if adversary knows communication graph
- [CCG+15]: "hidden graph" setup!
  - Each party sees only its neighborhood
  - Setup: Use a symmetric key infrastructure — every pair of parties decides if they have edge

Non-interactive!

Background: Ideas from [CCG+15]

Background: Ideas from [CCG+15]

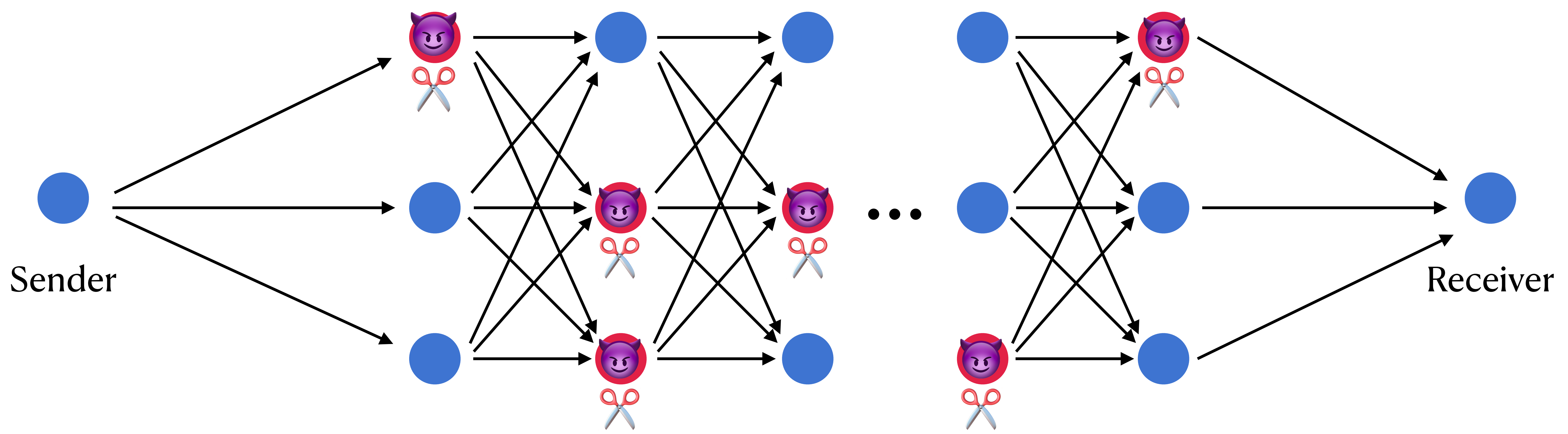
Reliable message transmission (RMT)



Background: Ideas from [CCG+15]

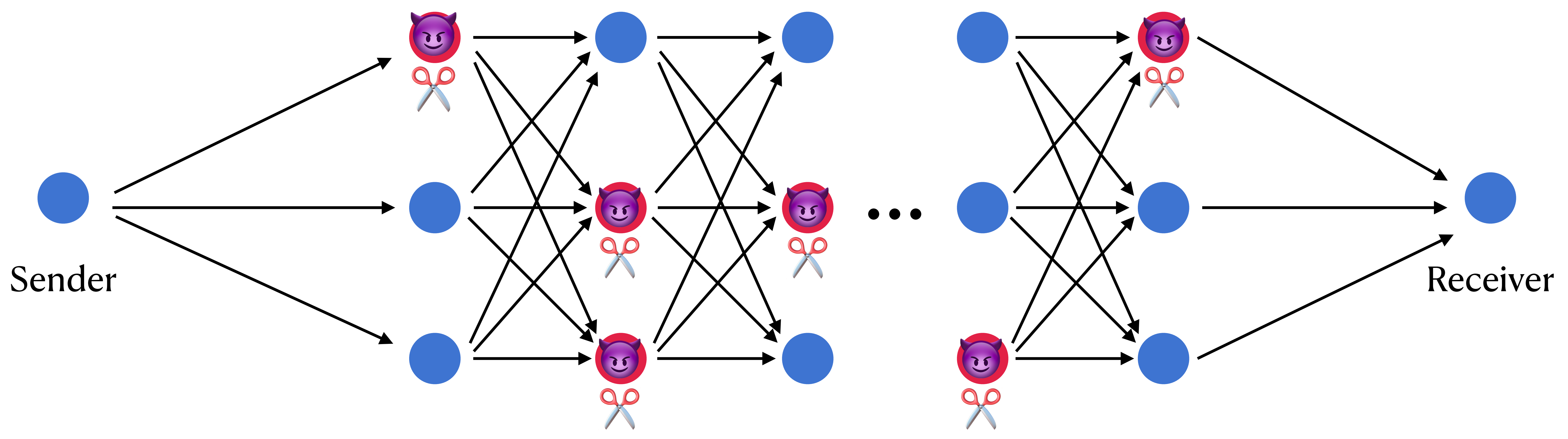
Background: Ideas from [CCG+15]

RMT over a hidden graph: “graph discovery game”



Background: Ideas from [CCG+15]

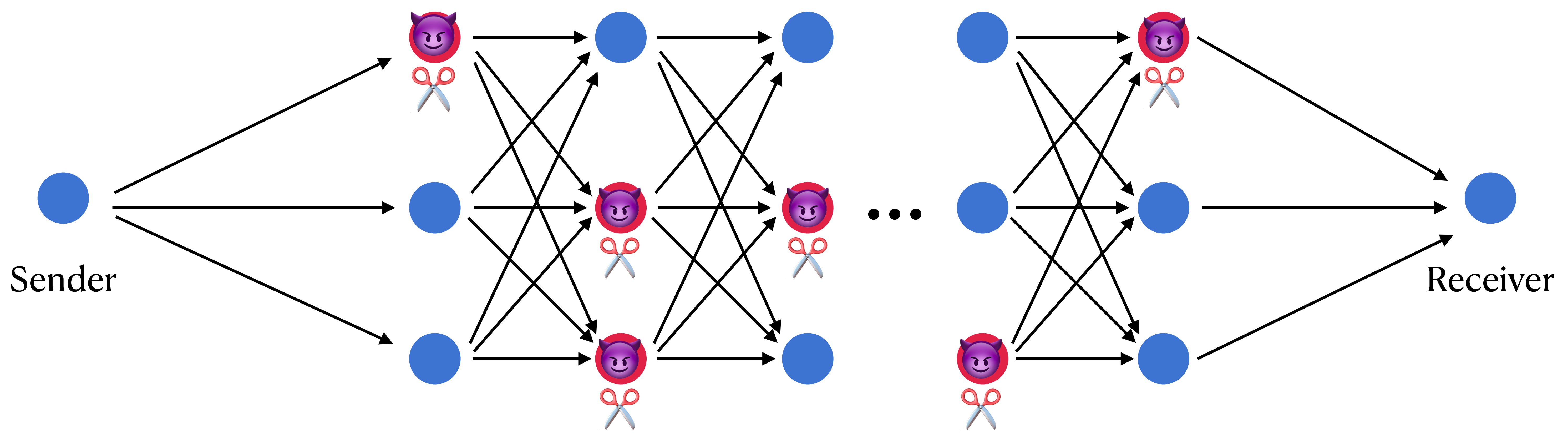
RMT over a hidden graph: "graph discovery game"



- CL RMT is a fundamental building block of CL MPC

Background: Ideas from [CCG+15]

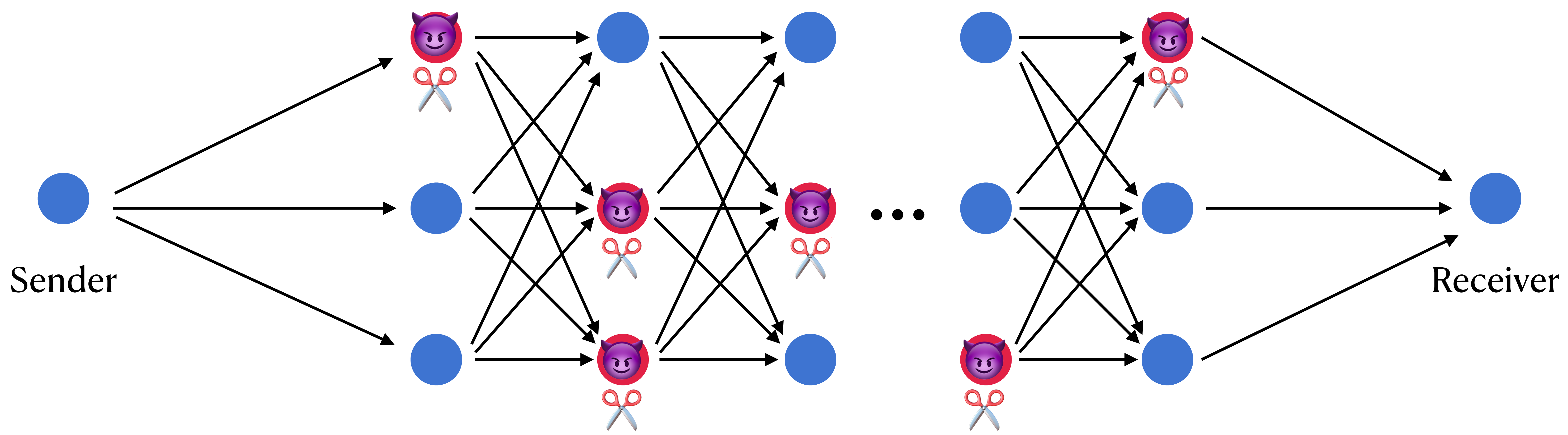
RMT over a hidden graph: "graph discovery game"



- CL RMT is a fundamental building block of CL MPC
- First prove results for CL RMT, then later extrapolate to CL MPC

Background: Ideas from [CCG+15]

RMT over a hidden graph: “graph discovery game”



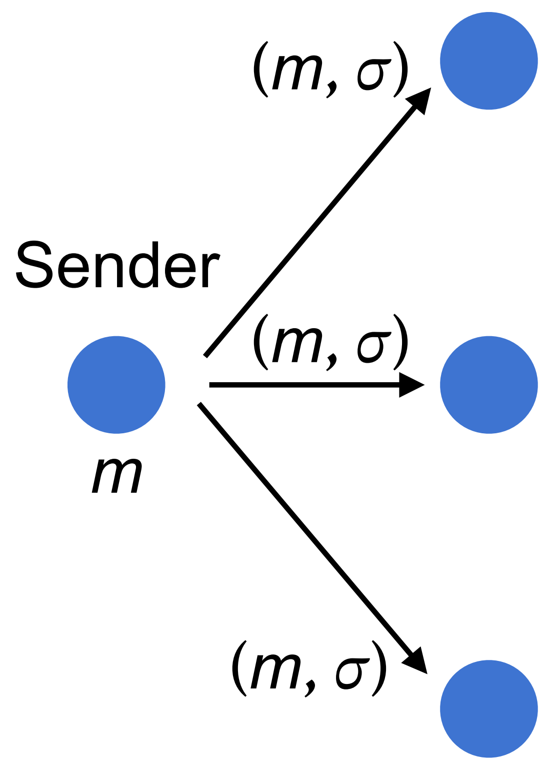
- CL RMT is a fundamental building block of CL MPC
- First prove results for CL RMT, then later extrapolate to CL MPC

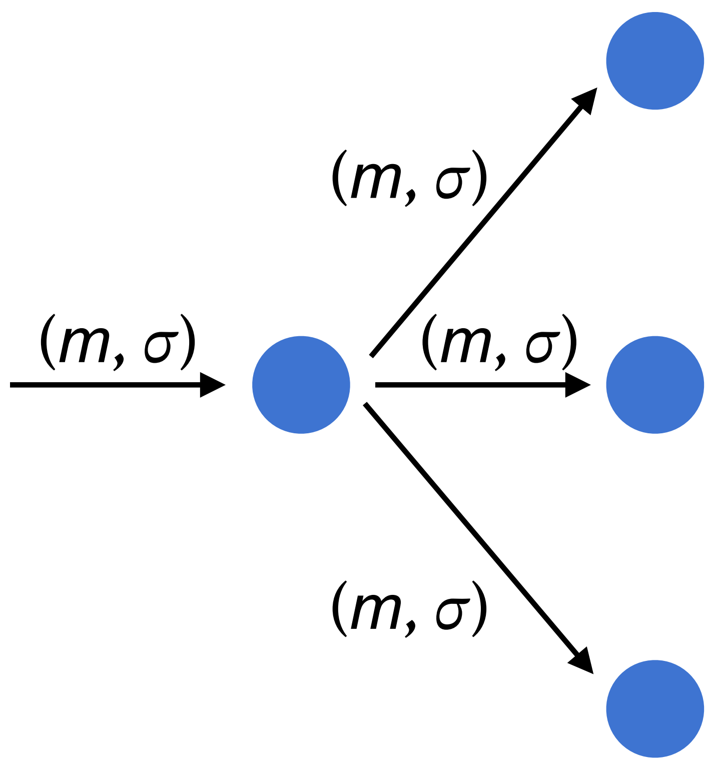
This talk

Communication-Local MPC

	Type of Adversary	Assumptions (on operations)			Result: CL MPC is...
		Atomic Multisend-and-Erase	Erasures	Atomic Multisend	
[BGT13]	Static	×	×	×	**feasible** for $t < (1/3 - \varepsilon)n$ corruptions
[CCG+15]	Adaptive	✓	✓	✓	**feasible** for $t < n/2$ corruptions
This work	Adaptive	×	×	✓	**impossible** for linear corruption, using “store-and-forward” protocols
	Adaptive	×	✓	×	**feasible** for $t < (1/2 - \varepsilon)n$ corruptions (weak cryptographic assumptions)
	Adaptive	×	✓	×	**impossible** for $t > (1/2 + \varepsilon)n$ corruptions, using “store-and-forward” protocols
	Adaptive	×	×	×	**feasible** for $t < n/2$ corruptions (strong cryptographic assumptions)

Store-and-Forward (SF) Protocols





Impossibility of SF RMT without erasures

Impossibility of SF RMT without erasures

Theorem: Without erasures, there is no store-and-forward protocol for CL RMT between all pairs of parties, tolerating an adaptive adversary corrupting a constant fraction of parties.

Impossibility of SF RMT without erasures

Theorem: Without erasures, there is no store-and-forward protocol for CL RMT between all pairs of parties, tolerating an adaptive adversary corrupting a constant fraction of parties.

Proof sketch:

Impossibility of SF RMT without erasures

Theorem: Without erasures, there is no store-and-forward protocol for CL RMT between all pairs of parties, tolerating an adaptive adversary corrupting a constant fraction of parties.

Proof sketch:

- There is a sender-receiver pair separated by distance $> \ell = O(\log n / \log \log n)$

Impossibility of SF RMT without erasures

Theorem: Without erasures, there is no store-and-forward protocol for CL RMT between all pairs of parties, tolerating an adaptive adversary corrupting a constant fraction of parties.

Proof sketch:

- There is a sender-receiver pair separated by distance $> \ell = O(\log n / \log \log n)$
- **Adversarial strategy**: When (m, σ) travels ℓ hops away, randomly corrupt some constant fraction of parties

Impossibility of SF RMT without erasures

Theorem: Without erasures, there is no store-and-forward protocol for CL RMT between all pairs of parties, tolerating an adaptive adversary corrupting a constant fraction of parties.

Proof sketch:

- There is a sender-receiver pair separated by distance $> \ell = O(\log n / \log \log n)$
- **Adversarial strategy**: When (m, σ) travels ℓ hops away, randomly corrupt some constant fraction of parties
- At least one party gets corrupted in each neighbor's subgraph, w.h.p.

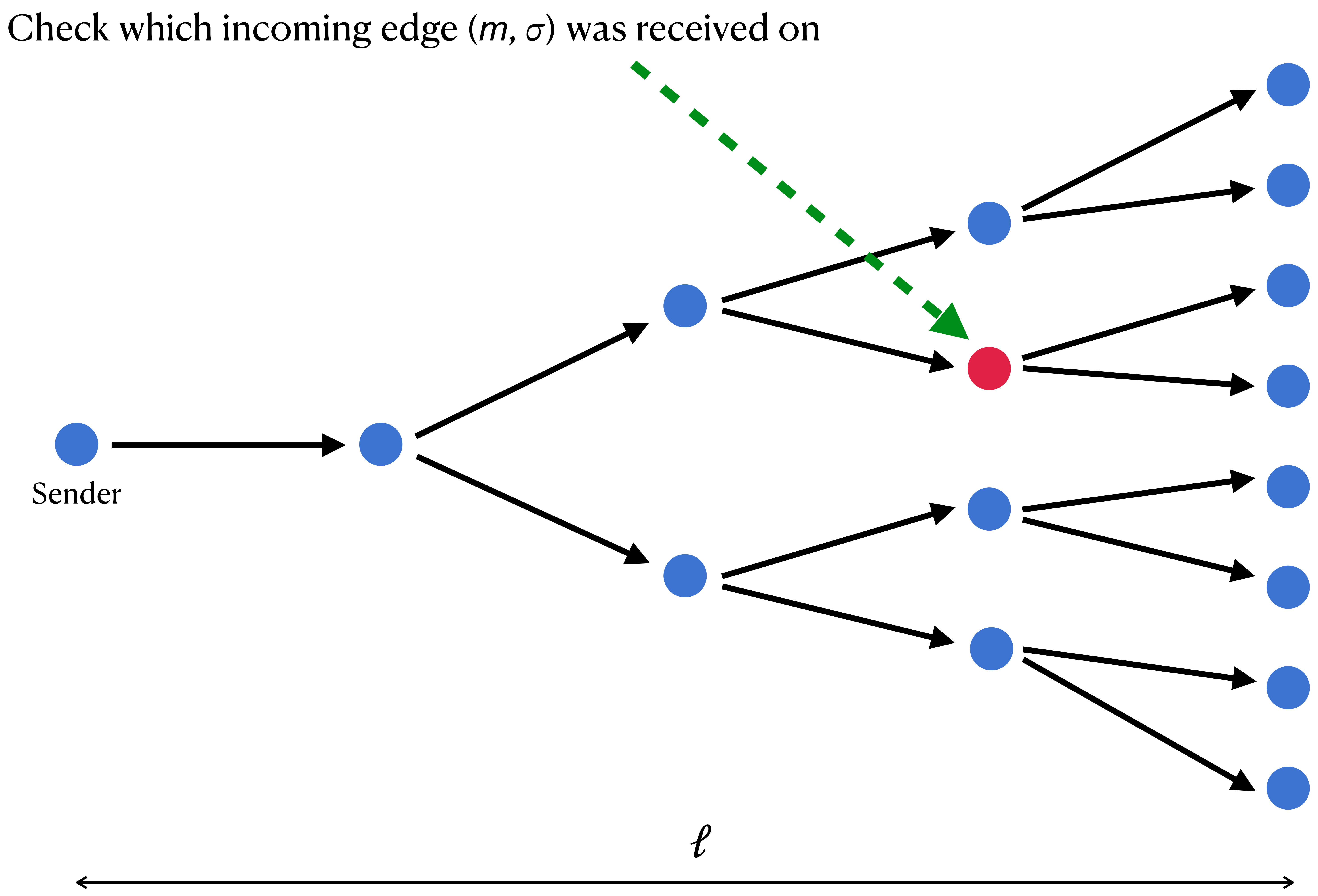
Impossibility of SF RMT without erasures

Theorem: Without erasures, there is no store-and-forward protocol for CL RMT between all pairs of parties, tolerating an adaptive adversary corrupting a constant fraction of parties.

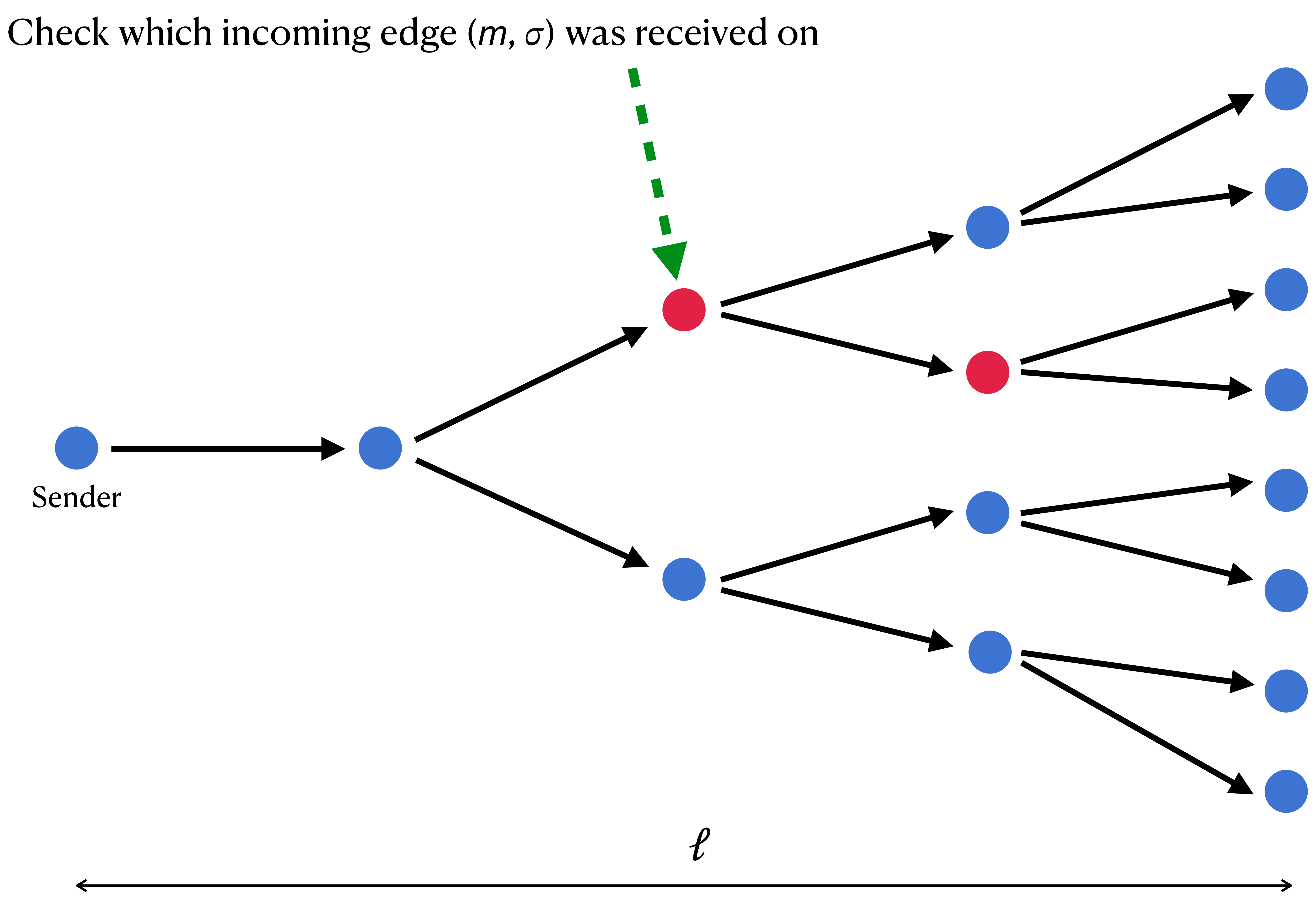
Proof sketch:

- There is a sender-receiver pair separated by distance $> \ell = O(\log n / \log \log n)$
- **Adversarial strategy**: When (m, σ) travels ℓ hops away, randomly corrupt some constant fraction of parties
- At least one party gets corrupted in each neighbor's subgraph, w.h.p.
- Sufficient for adversary to block the transmission!

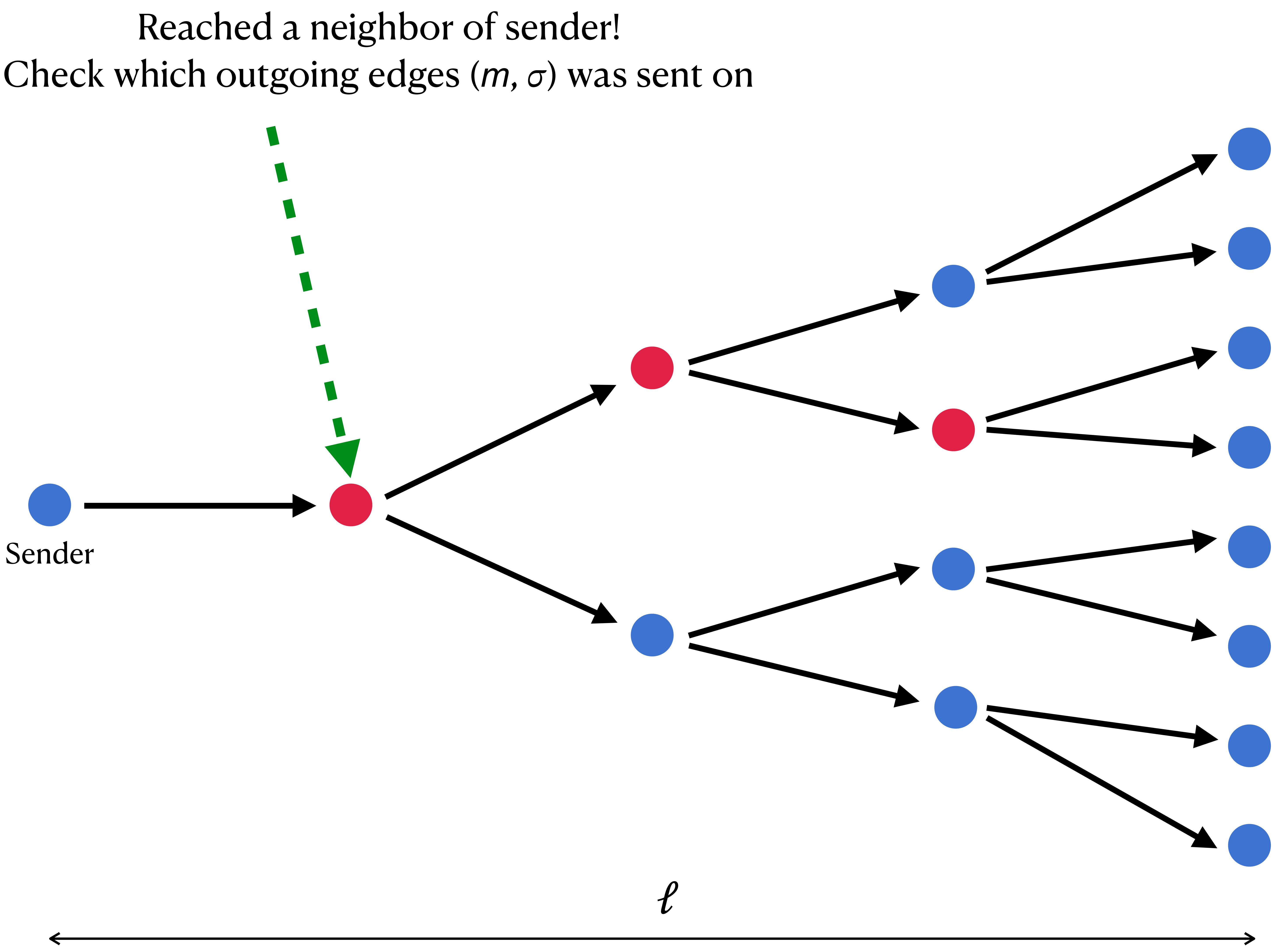
Impossibility of SF RMT without erasures



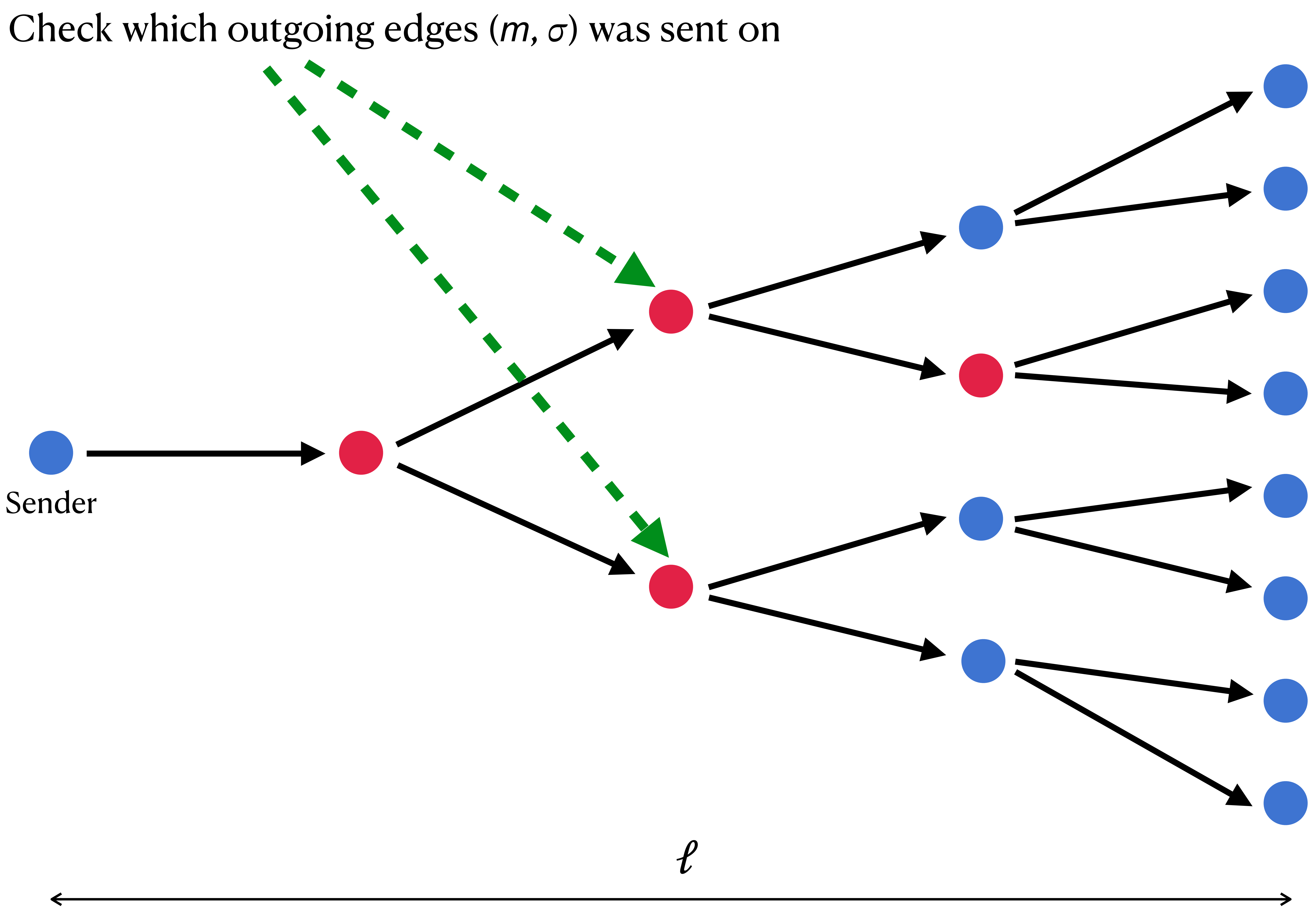
Impossibility of SF RMT without erasures



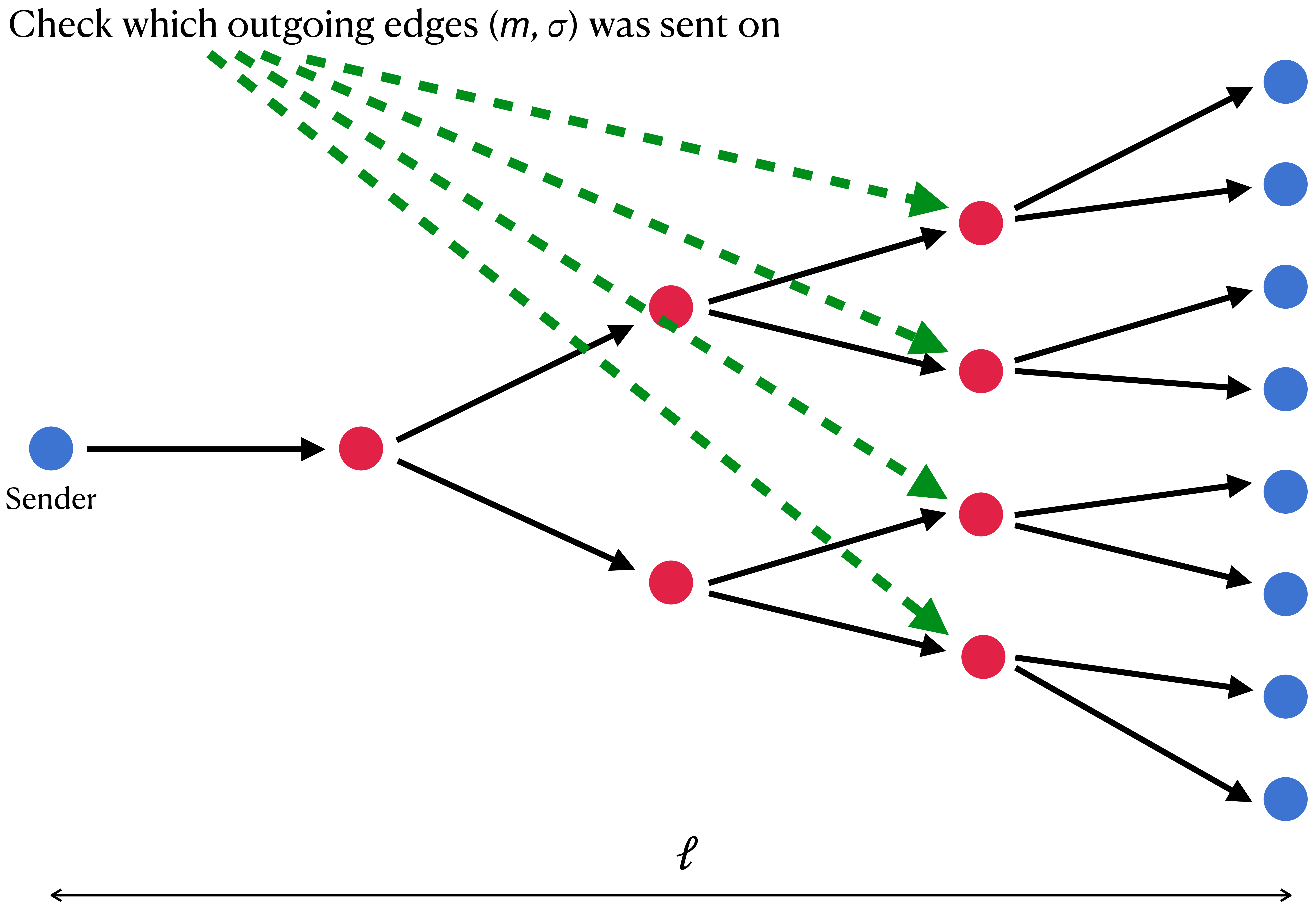
Impossibility of SF RMT without erasures



Impossibility of SF RMT without erasures



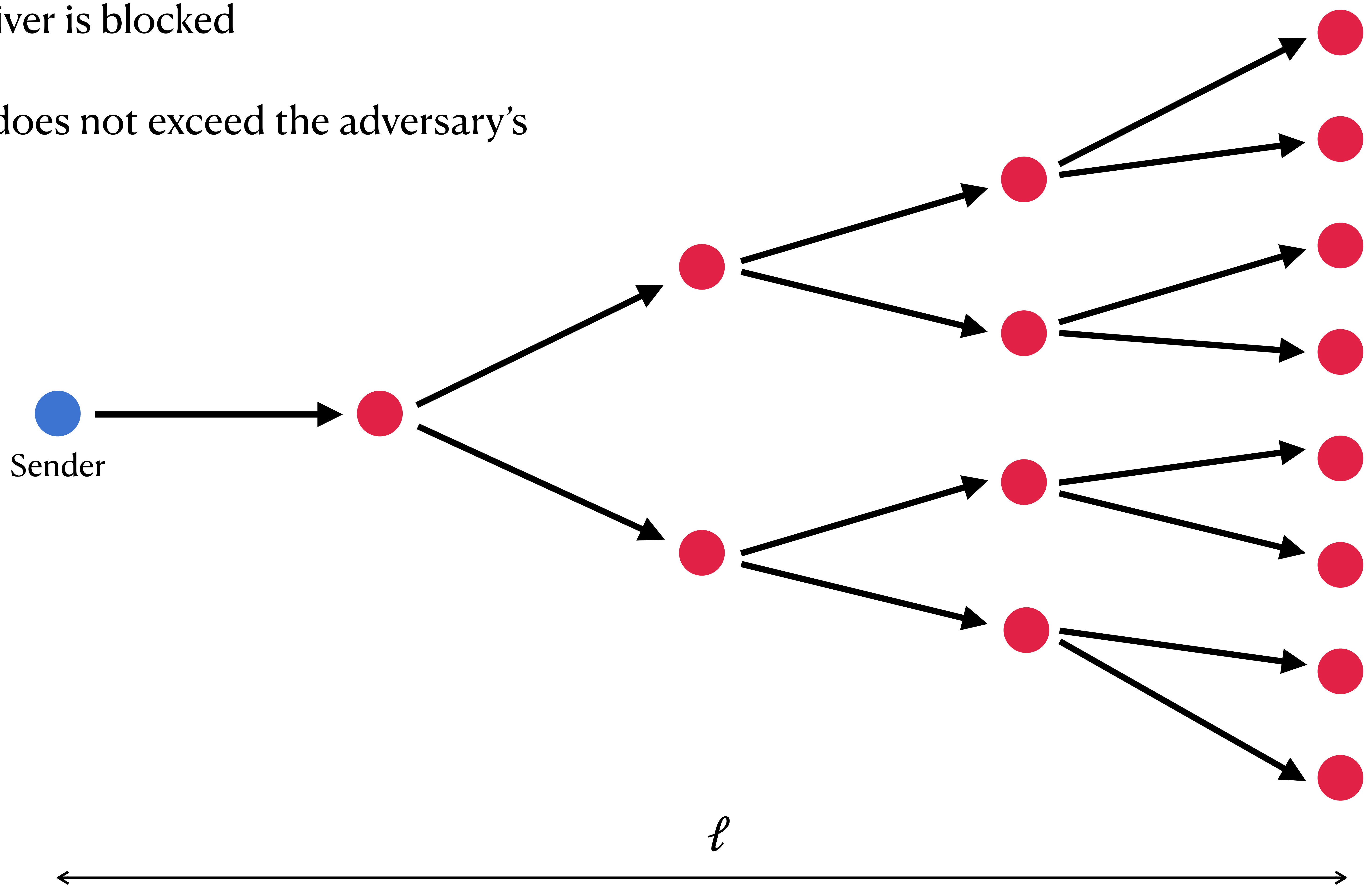
Impossibility of SF RMT without erasures



Impossibility of SF RMT without erasures

All parties (except sender) with (m, σ) are corrupted, and the transmission to receiver is blocked

Can be shown that this does not exceed the adversary's corruption budget



Communication-Local MPC

	Type of Adversary	Assumptions (on operations)			Result: CL MPC is...
		Atomic Multisend-and-Erase	Erasures	Atomic Multisend	
[BGT13]	Static	×	×	×	**feasible** for $t < (1/3 - \varepsilon)n$ corruptions
[CCG+15]	Adaptive	✓	✓	✓	**feasible** for $t < n/2$ corruptions
This work	Adaptive	×	×	✓	**impossible** for linear corruption, using "store-and-forward" protocols
	Adaptive	×	✓	×	**feasible** for $t < (1/2 - \varepsilon)n$ corruptions (weak cryptographic assumptions)
	Adaptive	×	✓	×	**impossible** for $t > (1/2 + \varepsilon)n$ corruptions, using "store-and-forward" protocols
	Adaptive	×	×	×	**feasible** for $t < n/2$ corruptions (strong cryptographic assumptions)

Feasibility of RMT (without erasures/atomic multisend!)

Feasibility of RMT (without erasures/atomic multisend!)

Theorem: Assuming a PKI, hidden graph setup, trapdoor permutations with a reverse domain sampler, and compact and malicious circuit-private FHE [OPP14], there is a polylog(n)-round CL RMT protocol for a single pair of parties, tolerating adaptive corruption of $t \leq (1\text{-}\varepsilon)n$ parties.

Feasibility of RMT (without erasures/atomic multisend!)

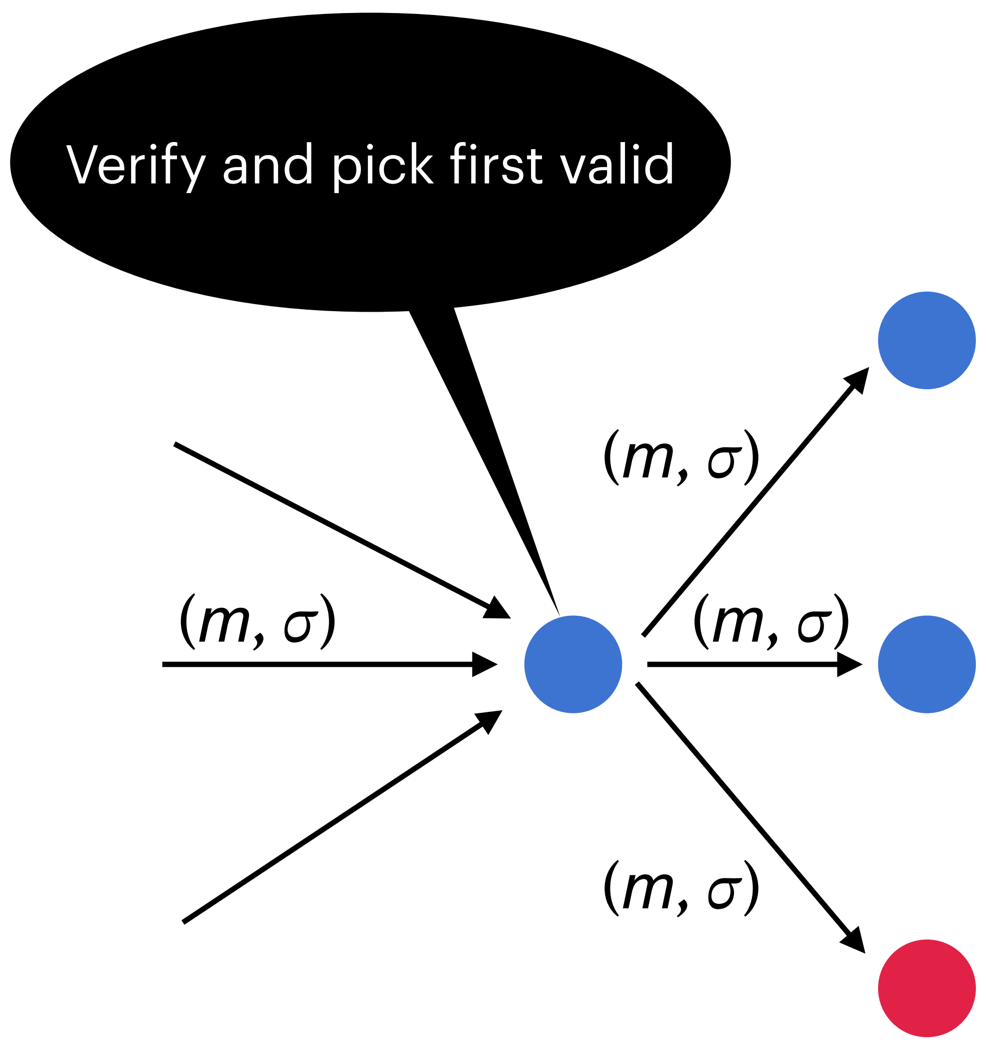
Theorem: Assuming a PKI, hidden graph setup, trapdoor permutations with a reverse domain sampler, and compact and malicious circuit-private FHE [OPP14], there is a polylog(n)-round CL RMT protocol for a single pair of parties, tolerating adaptive corruption of $t \leq (1\text{-}\varepsilon)n$ parties.

- Main insight: Hide the store-and-forward procedure under FHE!

Feasibility of RMT (without erasures/atomic multisend!)

Theorem: Assuming a PKI, hidden graph setup, trapdoor permutations with a reverse domain sampler, and compact and malicious circuit-private FHE [OPP14], there is a polylog(n)-round CL RMT protocol for a single pair of parties, tolerating adaptive corruption of $t \leq (1\text{-}\varepsilon)n$ parties.

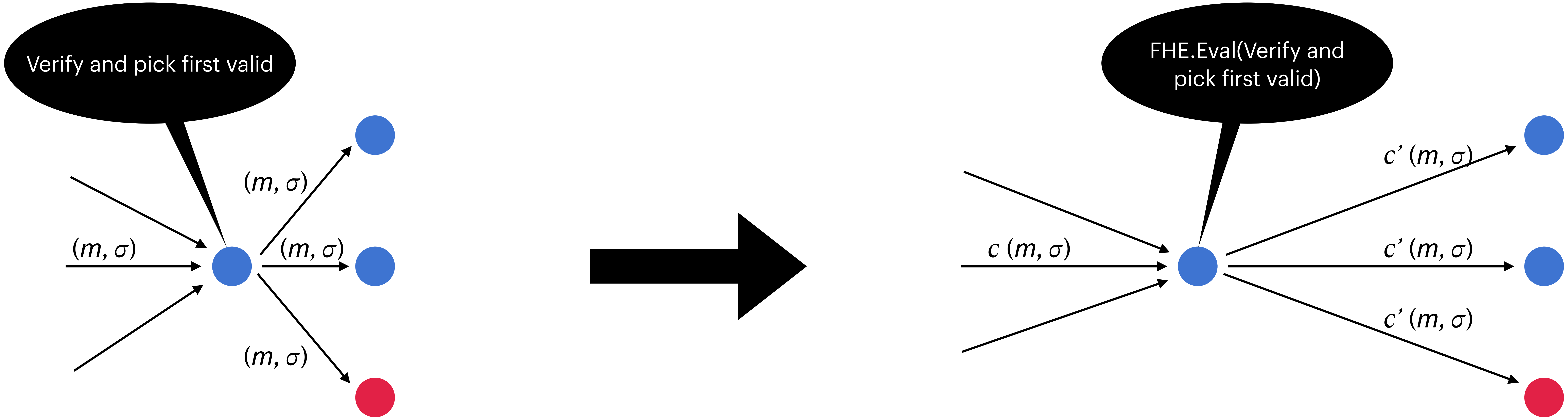
- Main insight: Hide the store-and-forward procedure under FHE!



Feasibility of RMT (without erasures/atomic multisend!)

Theorem: Assuming a PKI, hidden graph setup, trapdoor permutations with a reverse domain sampler, and compact and malicious circuit-private FHE [OPP14], there is a polylog(n)-round CL RMT protocol for a single pair of parties, tolerating adaptive corruption of $t \leq (1\text{-}\varepsilon)n$ parties.

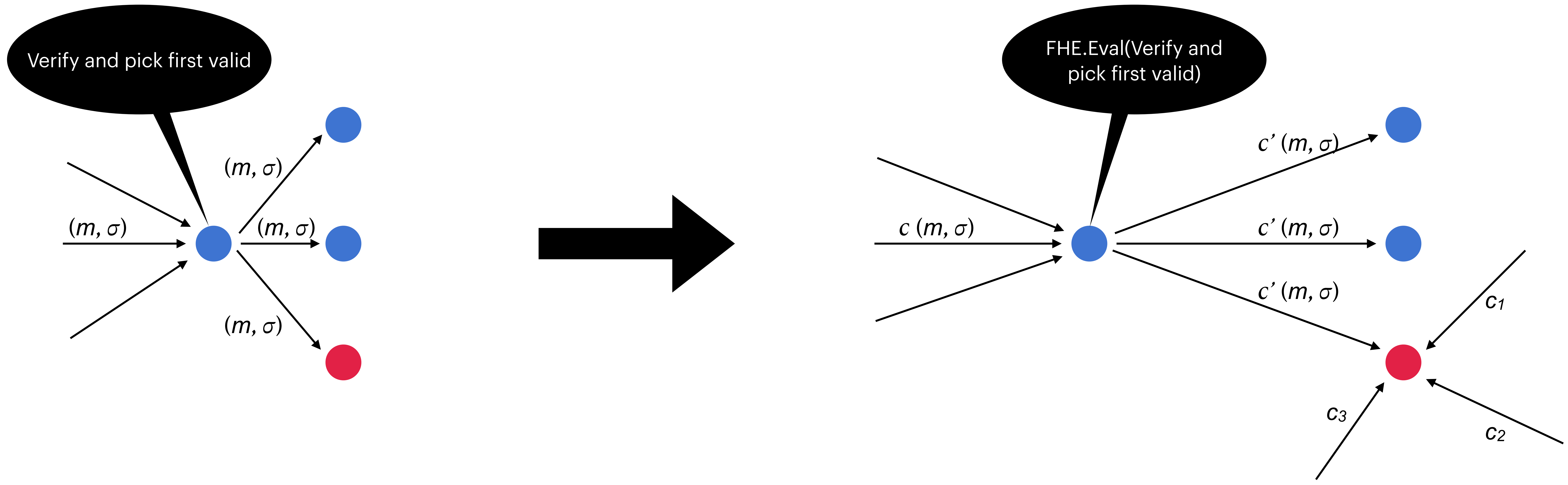
- Main insight: Hide the store-and-forward procedure under FHE!



Feasibility of RMT (without erasures/atomic multisend!)

Theorem: Assuming a PKI, hidden graph setup, trapdoor permutations with a reverse domain sampler, and compact and malicious circuit-private FHE [OPP14], there is a polylog(n)-round CL RMT protocol for a single pair of parties, tolerating adaptive corruption of $t \leq (1\text{-}\varepsilon)n$ parties.

- Main insight: Hide the store-and-forward procedure under FHE!



Feasibility of RMT (without erasures/atomic multisend!)

Feasibility of RMT (without erasures/atomic multisend!)

Several caveats:

Feasibility of RMT (without erasures/atomic multisend!)

Several caveats:

- Ciphertext size may reveal position of relayer in the hidden graph $\Rightarrow$ Compact FHE!

Feasibility of RMT (without erasures/atomic multisend!)

Several caveats:

- Ciphertext size may reveal position of relayer in the hidden graph $\Rightarrow$ Compact FHE!
- Adversarially-fed malicious ciphertexts may cause output of FHE evaluation to leak information $\Rightarrow$ Malicious circuit-private FHE!

Feasibility of RMT (without erasures/atomic multisend!)

Several caveats:

- Ciphertext size may reveal position of relayer in the hidden graph $\Rightarrow$ Compact FHE!
- Adversarially-fed malicious ciphertexts may cause output of FHE evaluation to leak information $\Rightarrow$ Malicious circuit-private FHE!
- [KTZ13]: Compact and adaptively secure FHE is impossible!

Feasibility of RMT (without erasures/atomic multisend!)

Several caveats:

- Ciphertext size may reveal position of relayer in the hidden graph $\Rightarrow$ Compact FHE!
- Adversarially-fed malicious ciphertexts may cause output of FHE evaluation to leak information $\Rightarrow$ Malicious circuit-private FHE!
- [KTZ13]: Compact and adaptively secure FHE is impossible!
  - Solution: Scheme for our specific function — verify message-signature pairs and select the first valid one

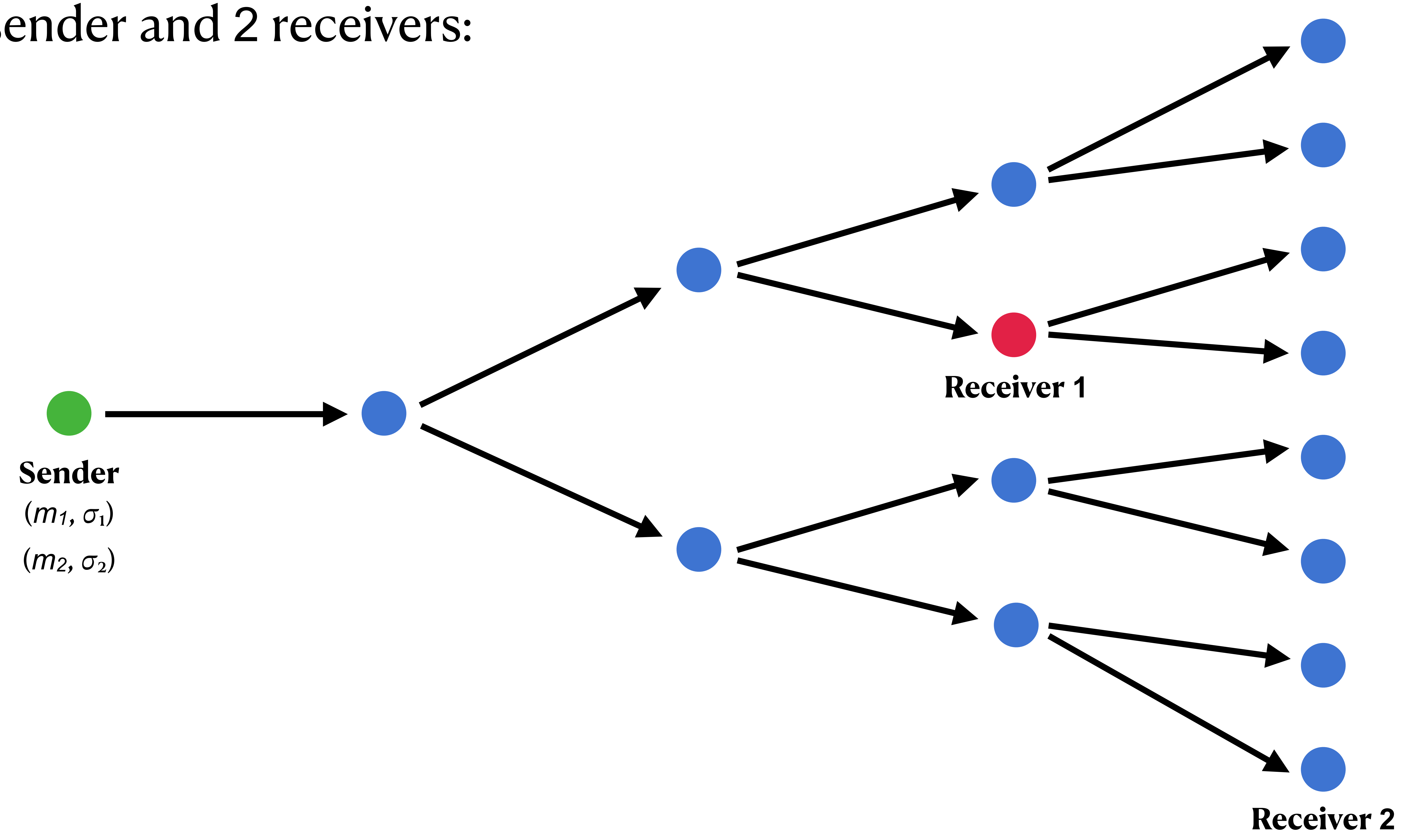
From single-pair RMT to all-to-all RMT

From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!

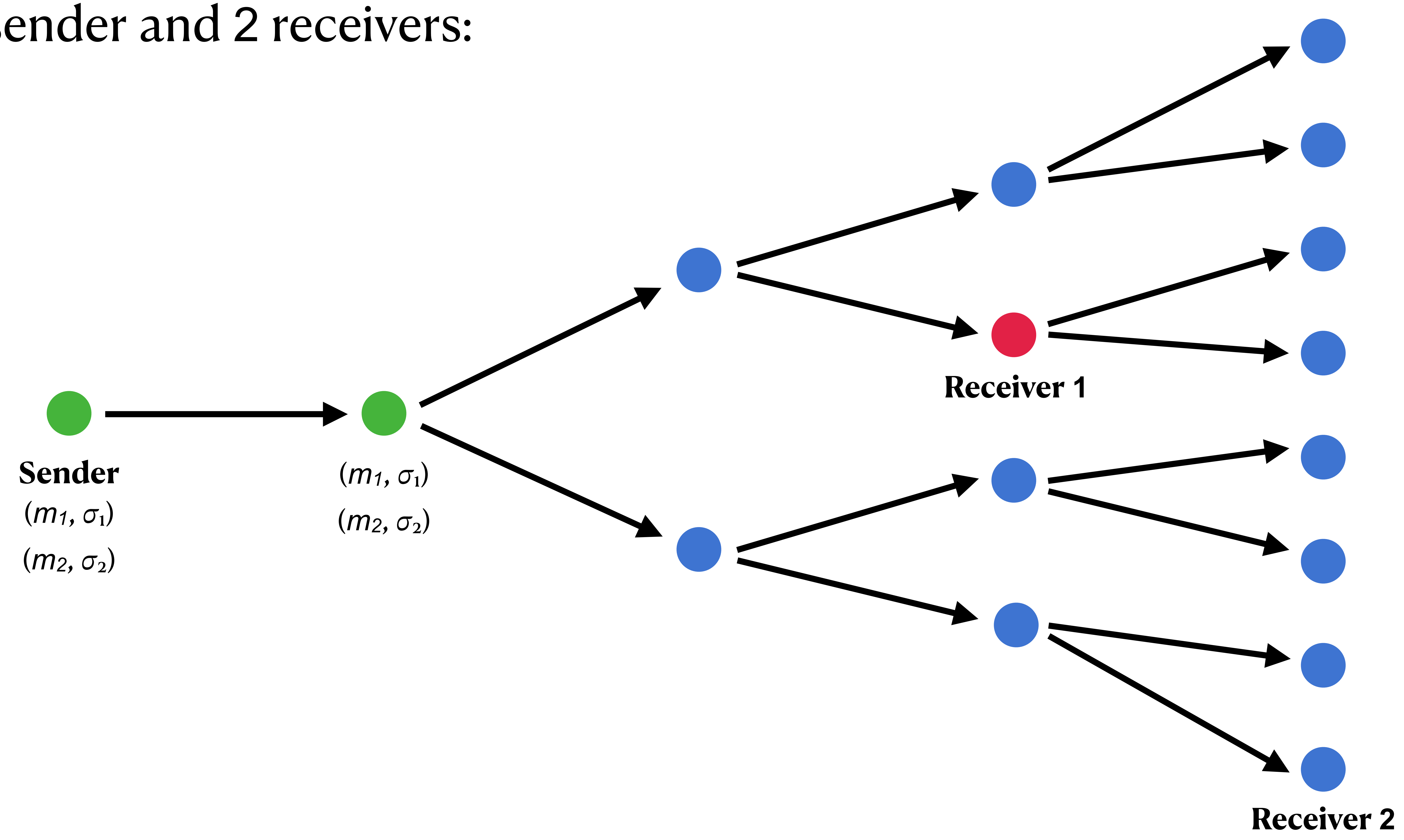
From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



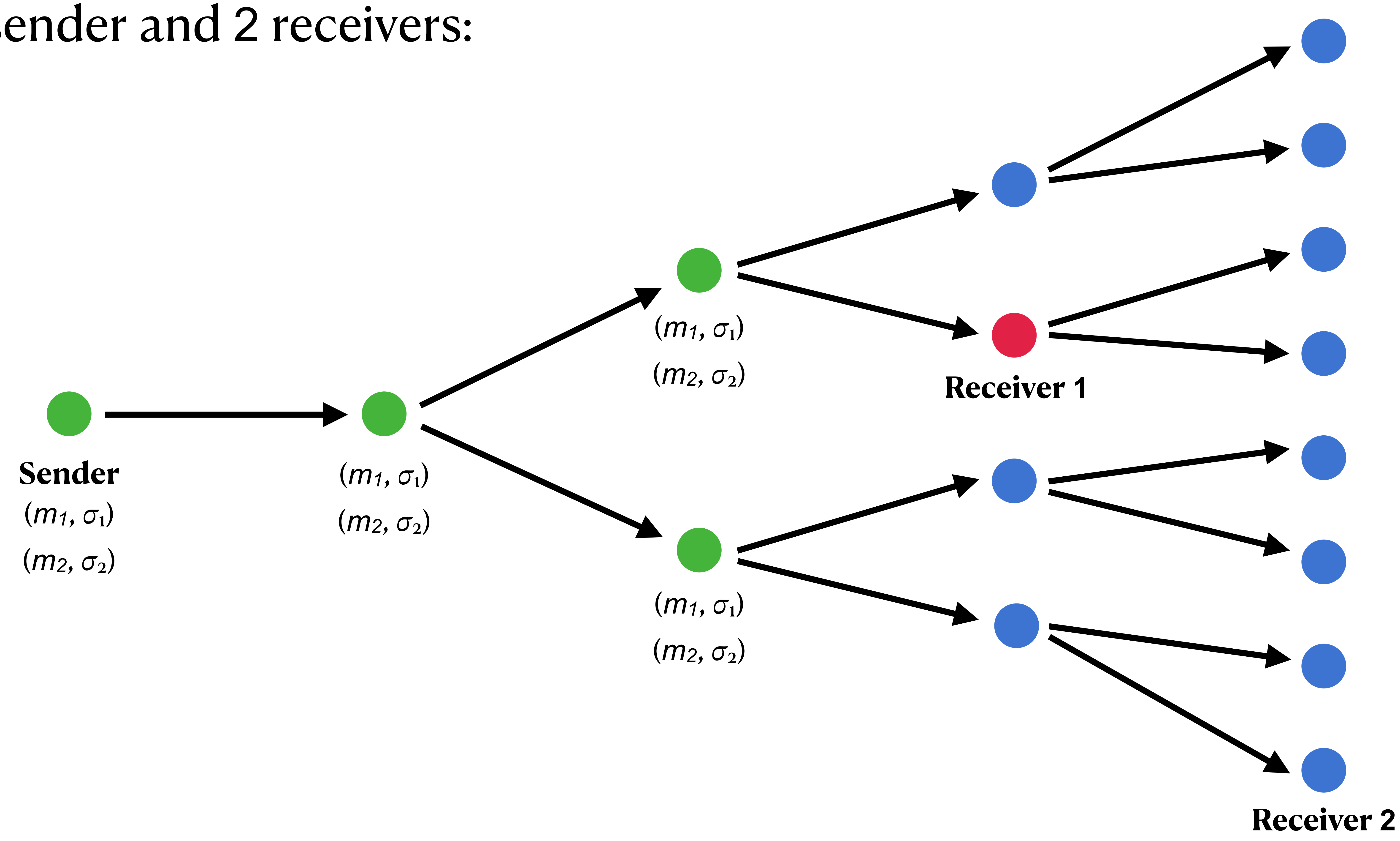
From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



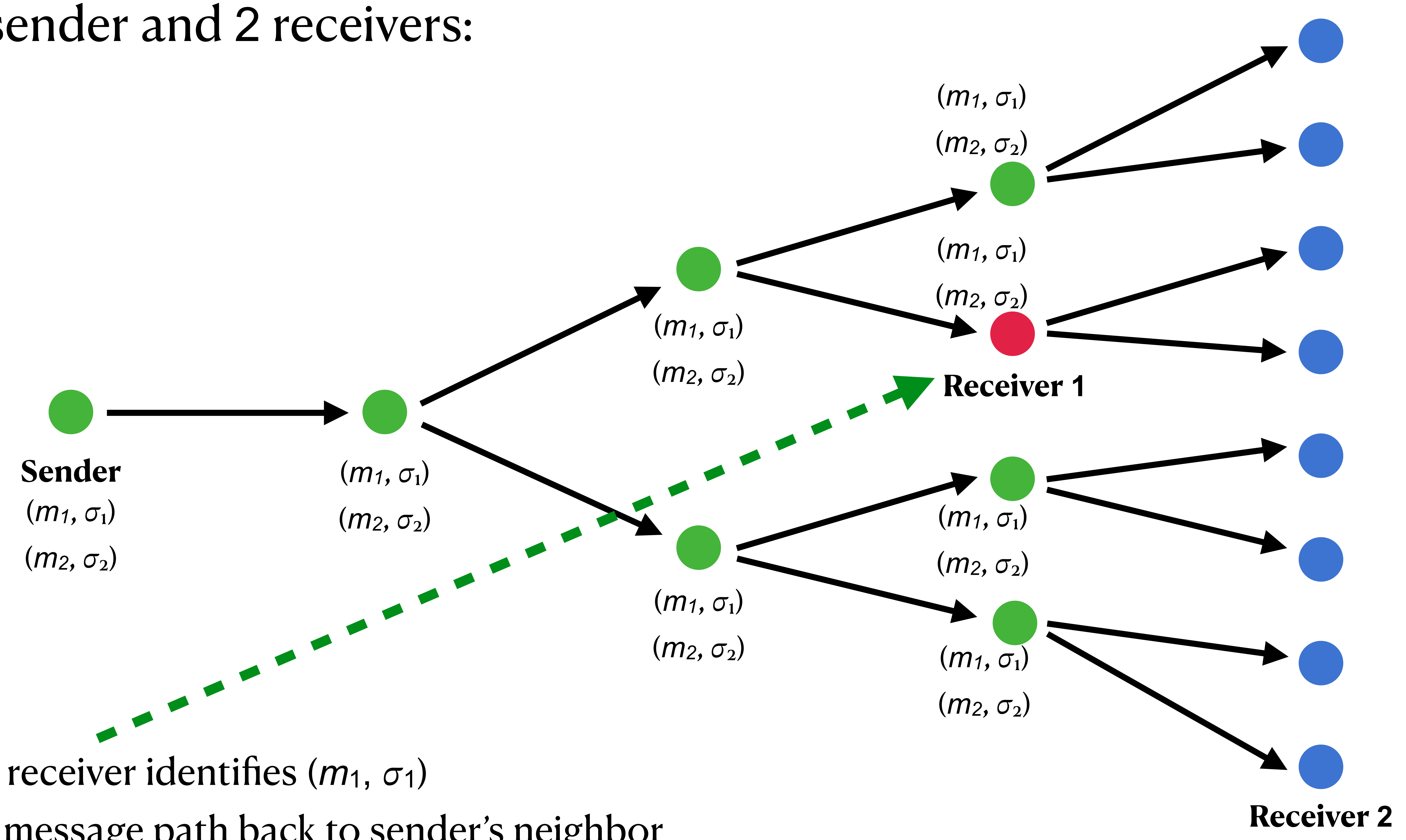
From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



From single-pair RMT to all-to-all RMT

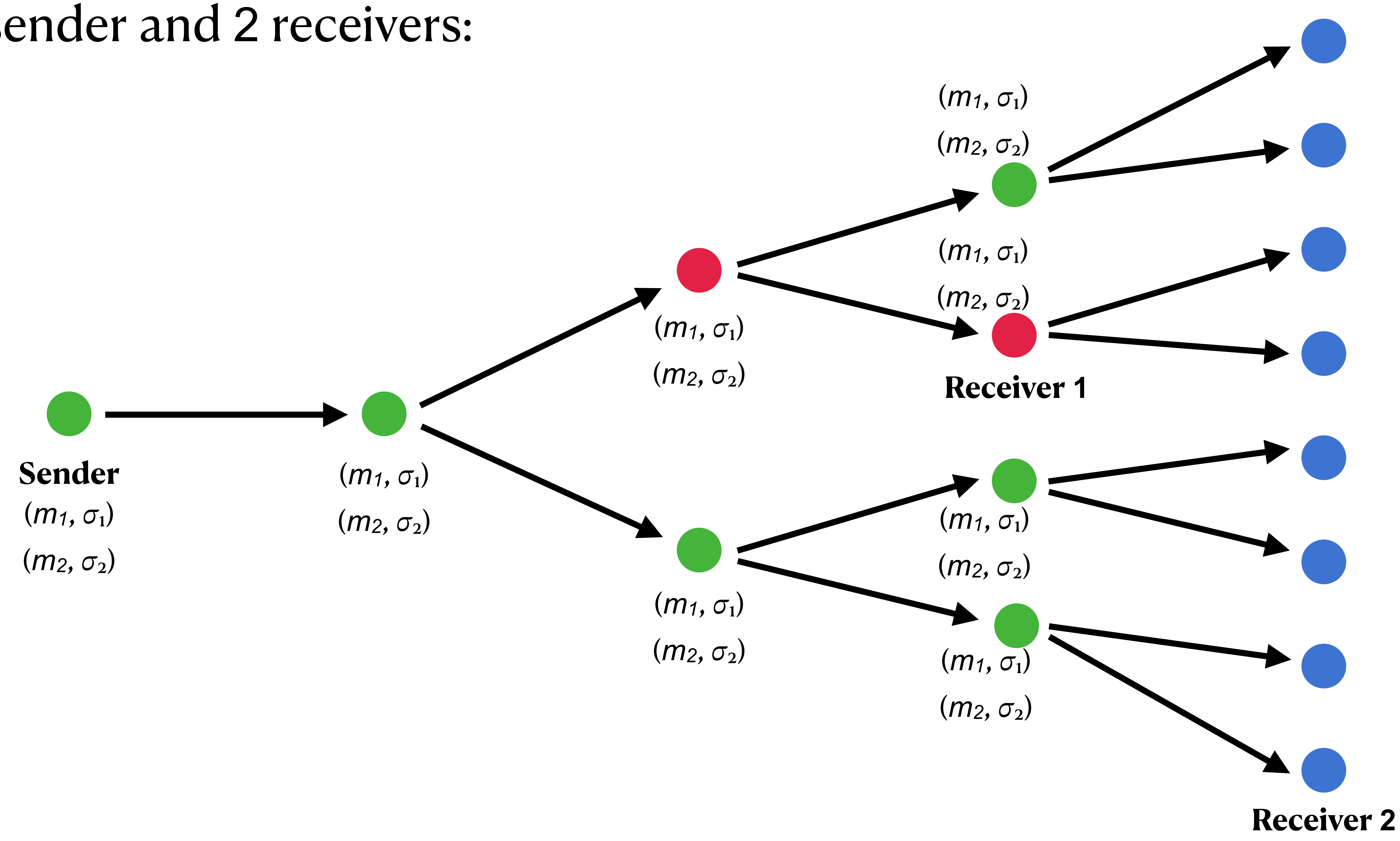
- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



Corrupted receiver identifies (m_1, σ_1)
Adversary can now trace message path back to sender's neighbor

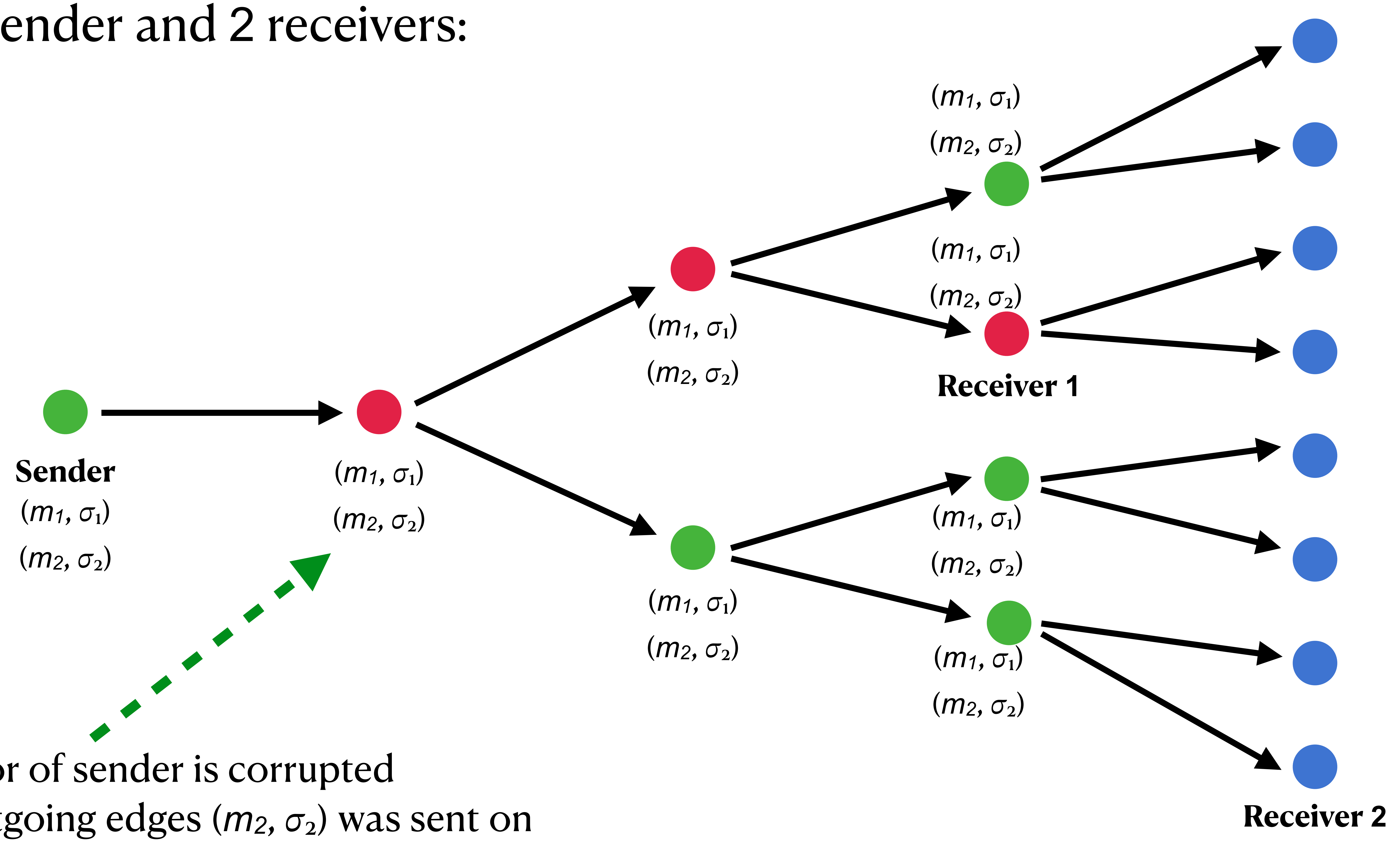
From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



From single-pair RMT to all-to-all RMT

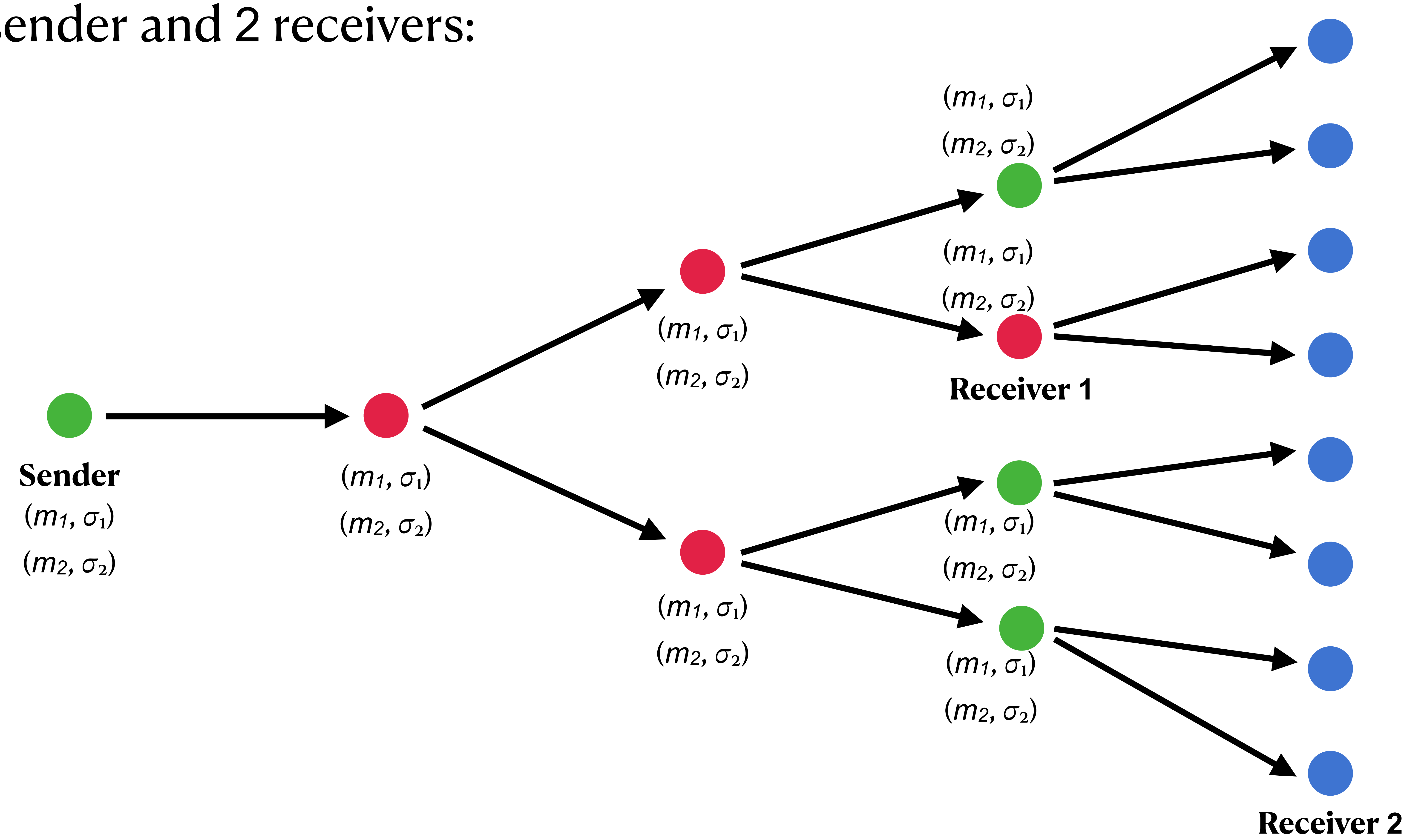
- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



Neighbor of sender is corrupted
Check which outgoing edges (m_2, σ_2) was sent on

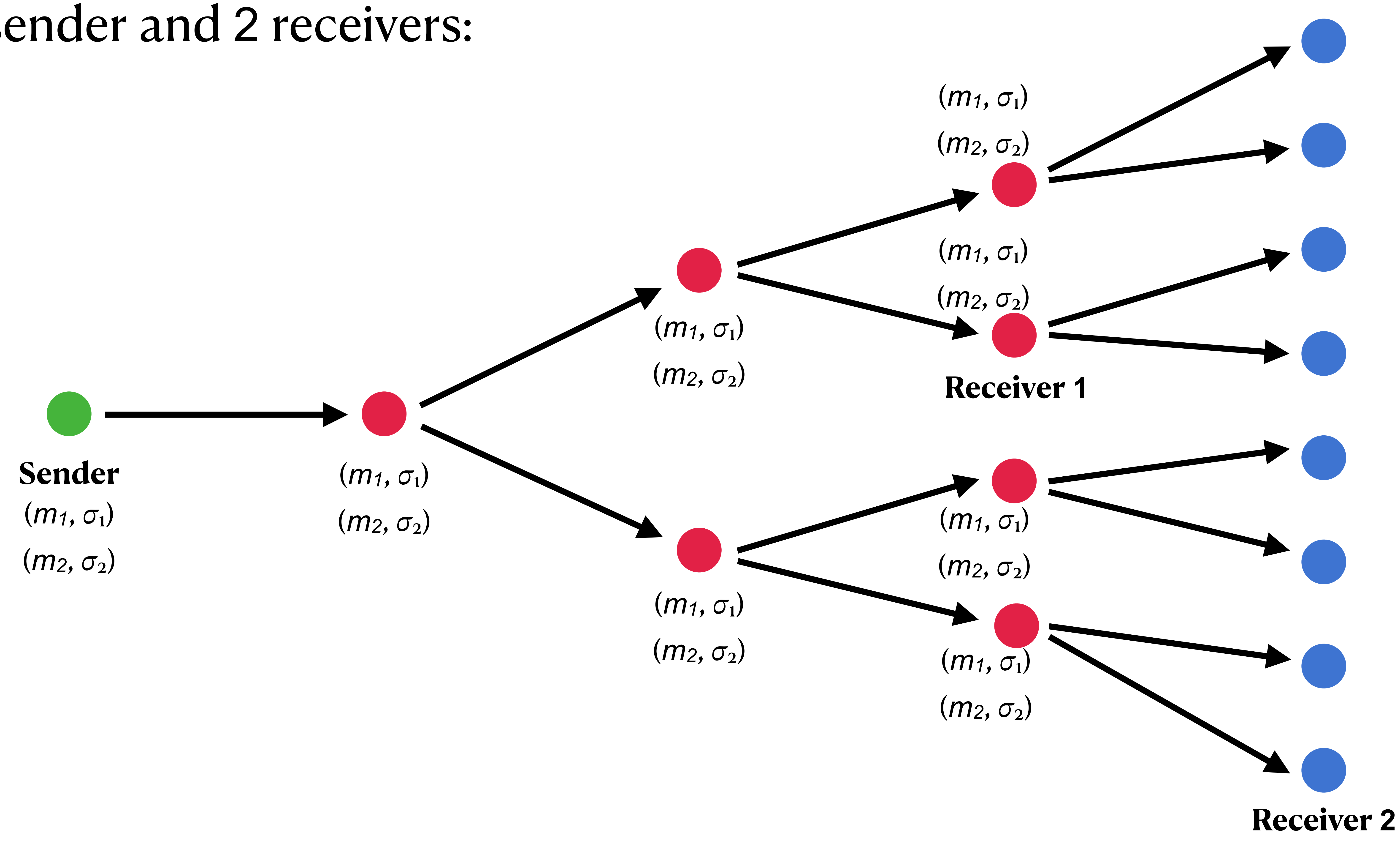
From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



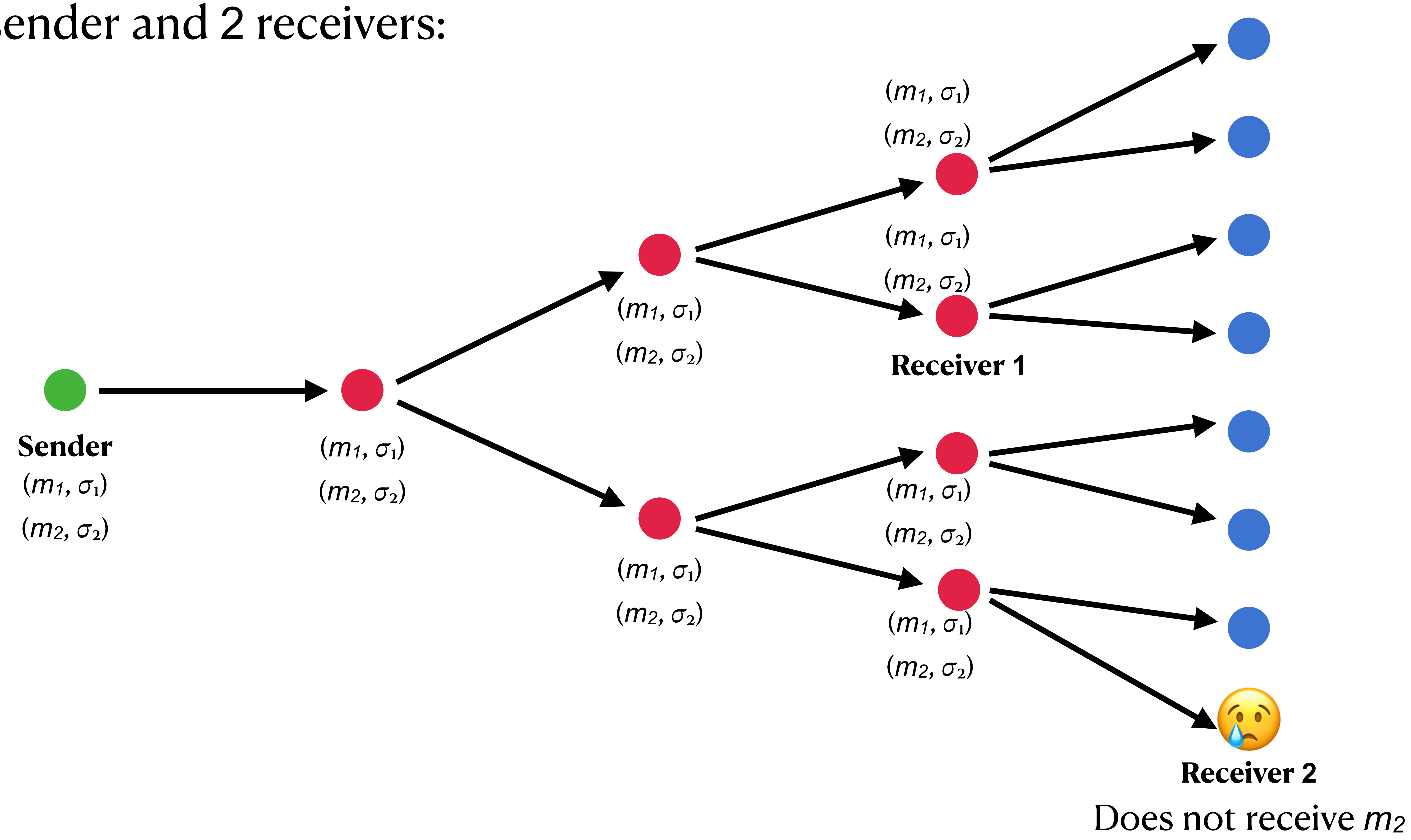
From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



From single-pair RMT to all-to-all RMT

- Parallel composition? Does not work!
- Example with 1 sender and 2 receivers:



From single-pair RMT to all-to-polylog(n) RMT

From single-pair RMT to all-to-polylog(n) RMT

- Best we can do is n senders and only polylog(n) receivers

From single-pair RMT to all-to-polylog(n) RMT

- Best we can do is n senders and only polylog(n) receivers
  - Use an independent hidden graph for each receiver

From single-pair RMT to all-to-polylog(n) RMT

- Best we can do is n senders and only polylog(n) receivers
  - Use an independent hidden graph for each receiver
  - Adversary cannot interfere with hidden graphs of honest receivers

From single-pair RMT to all-to-polylog(n) RMT

- Best we can do is n senders and only polylog(n) receivers
  - Use an independent hidden graph for each receiver
  - Adversary cannot interfere with hidden graphs of honest receivers
- We call this **sublinear output set (SOS-)RMT**

From single-pair RMT to all-to-polylog(n) RMT

- Best we can do is n senders and only polylog(n) receivers
  - Use an independent hidden graph for each receiver
  - Adversary cannot interfere with hidden graphs of honest receivers
- We call this **sublinear output set (SOS-)RMT**
- SOS-RMT can be used to achieve SOS-MPC

Open problems

- All-to-all RMT (without erasures)
- RMT (without erasures) from weaker cryptographic assumptions
- RMT with asynchronous communication (work in progress)

Thank you!

ePrint: 2024/1489