Privacy at the communication layer

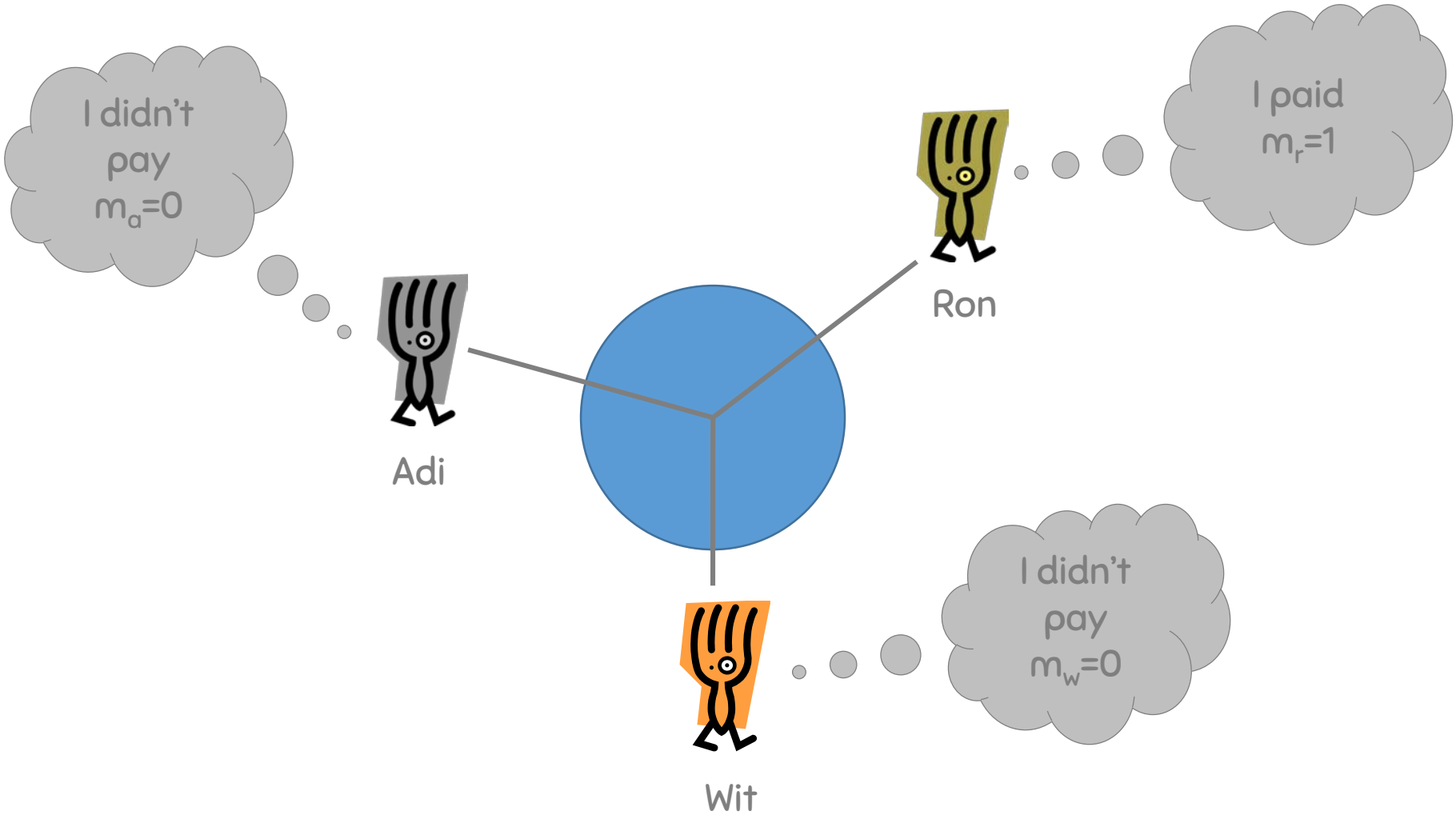
The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability
David Chaum 1988

CS-721

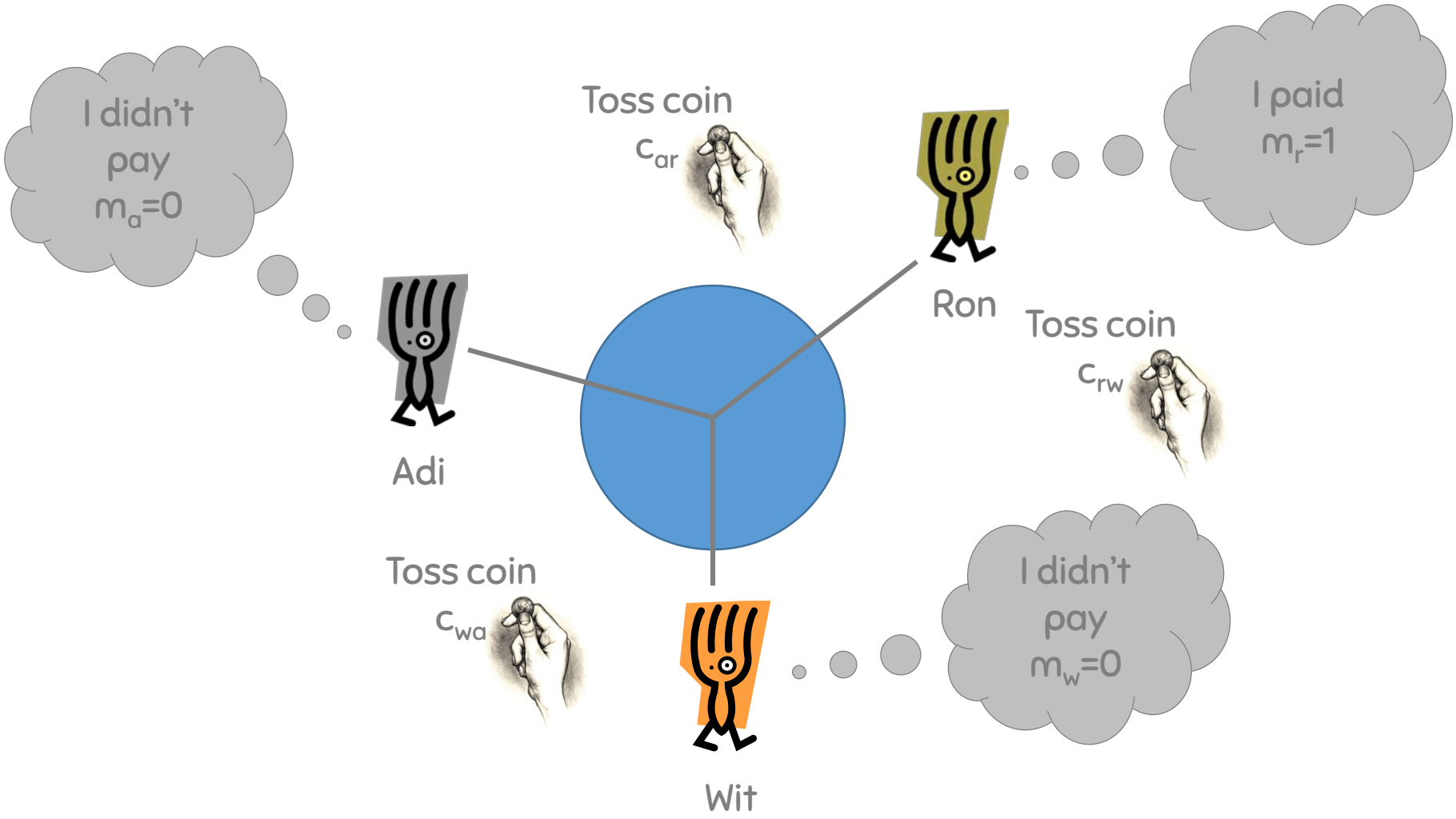
Carmela Troncoso
http://carmelatroncoso.com/

(borrowed slides from G. Danezis)

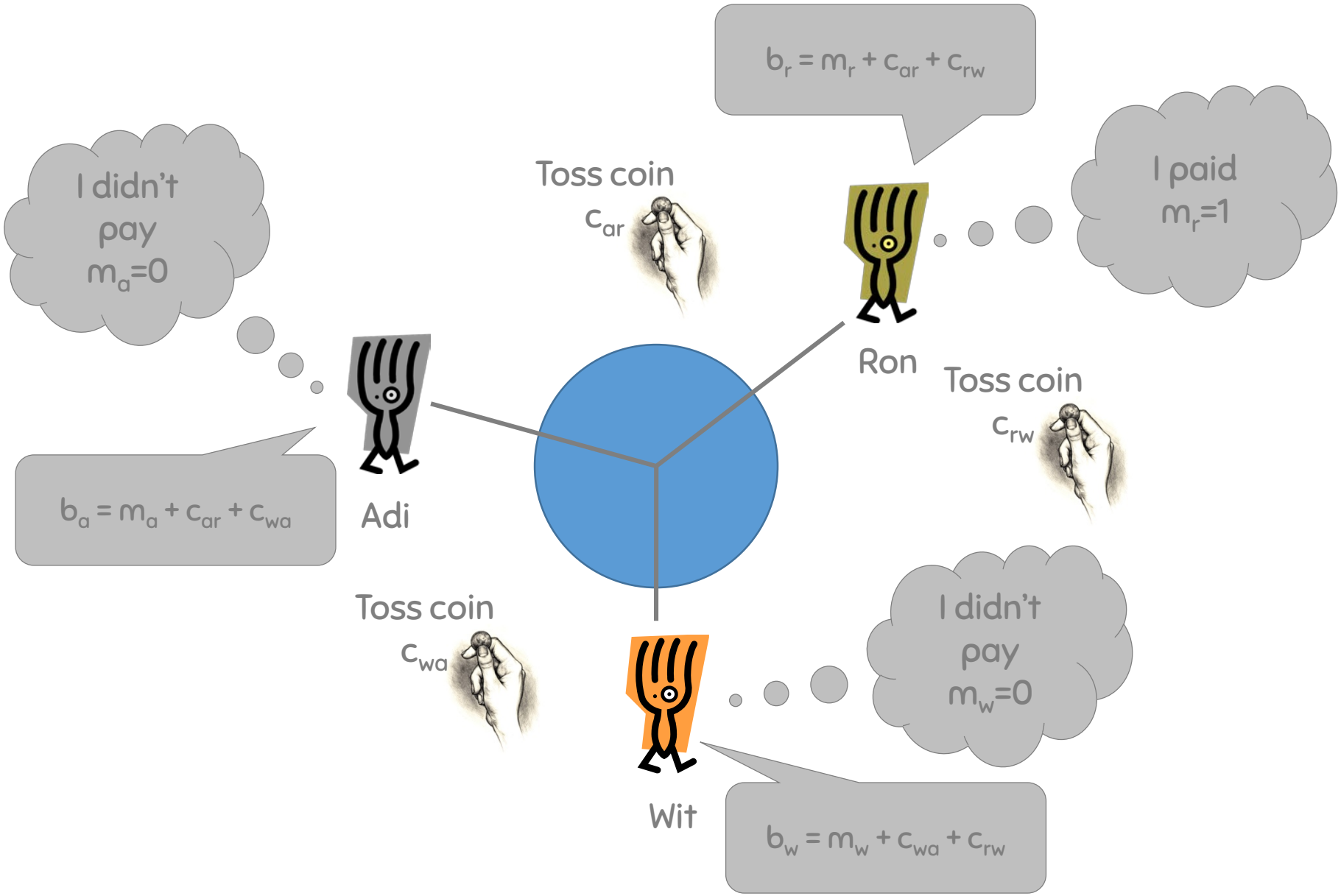
The Dining Cryptographers – Did the NSA pay?



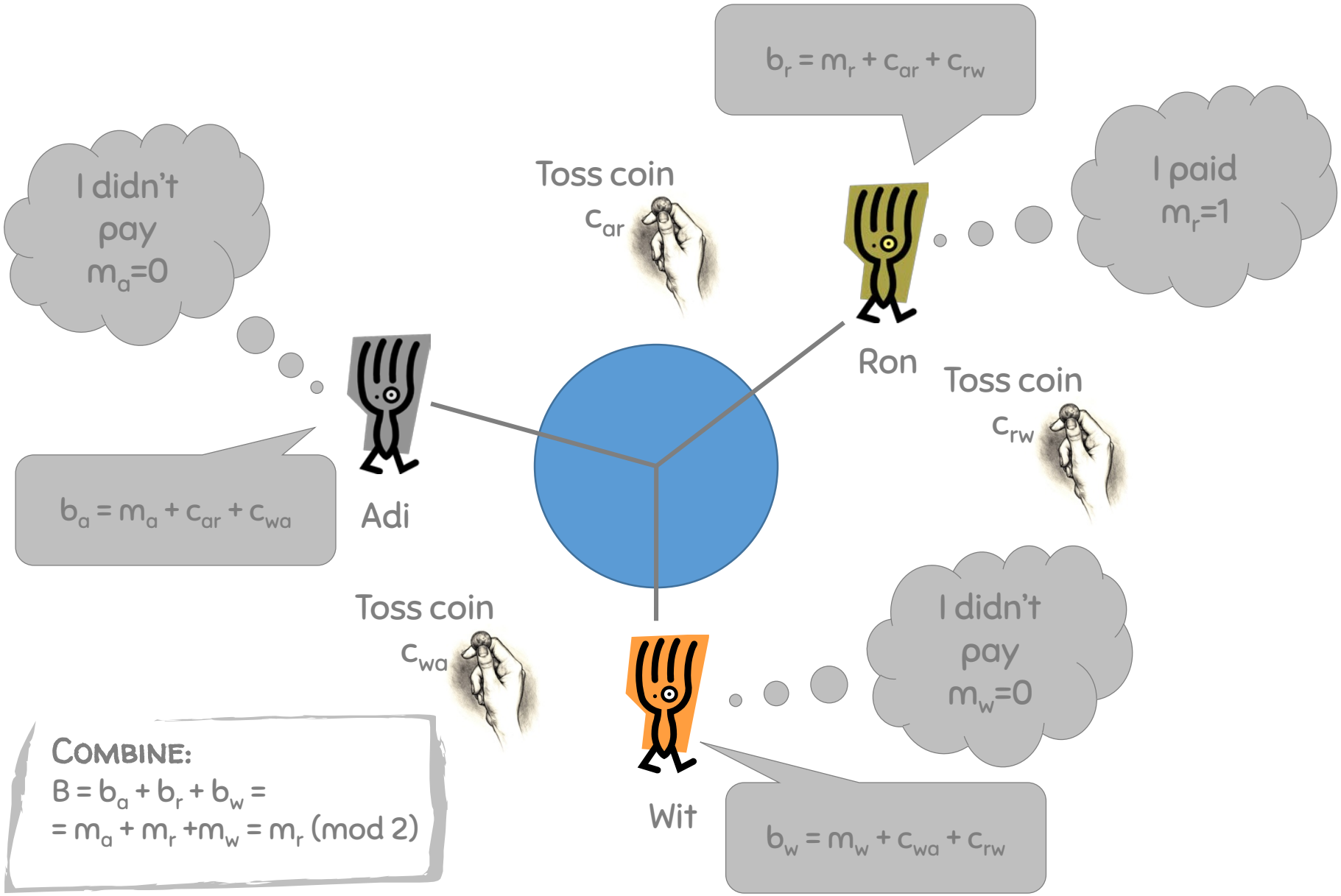
The Dining Cryptographers – Did the NSA pay?



The Dining Cryptographers – Did the NSA pay?

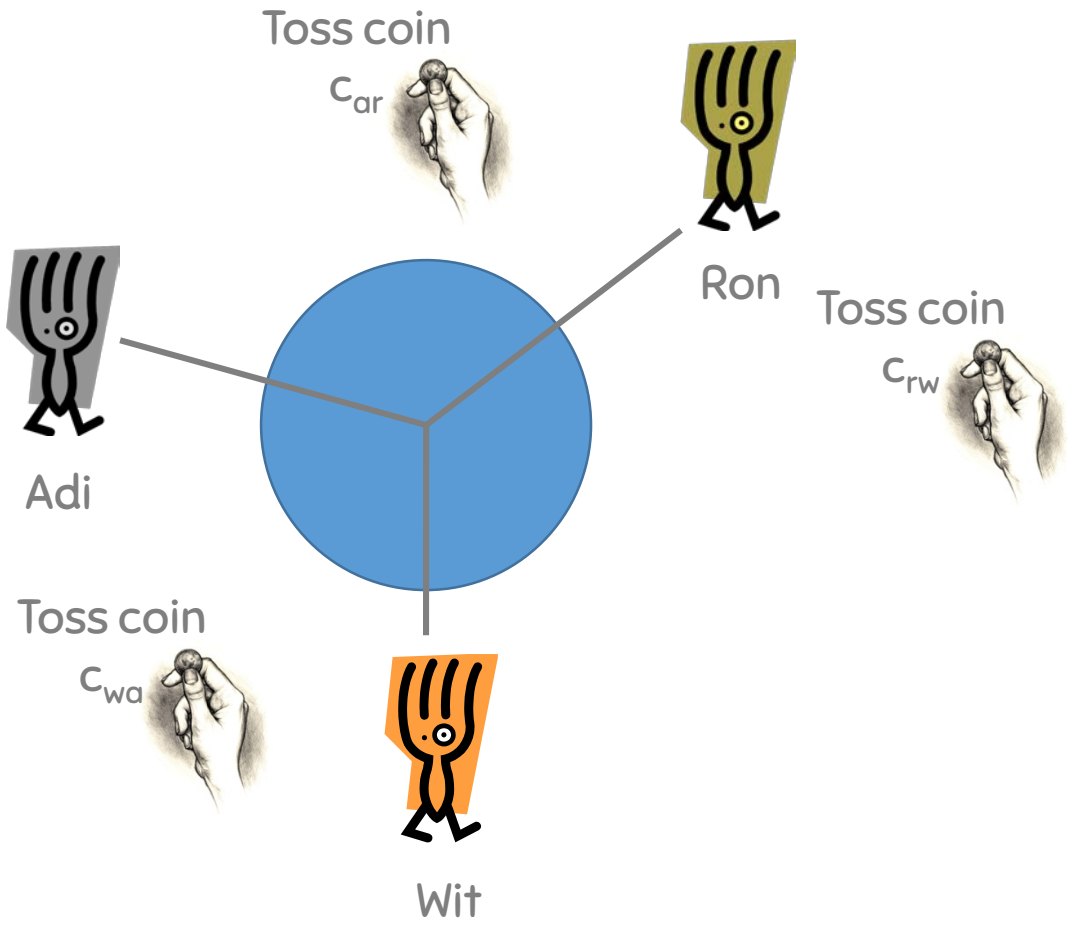


The Dining Cryptographers – Did the NSA pay?



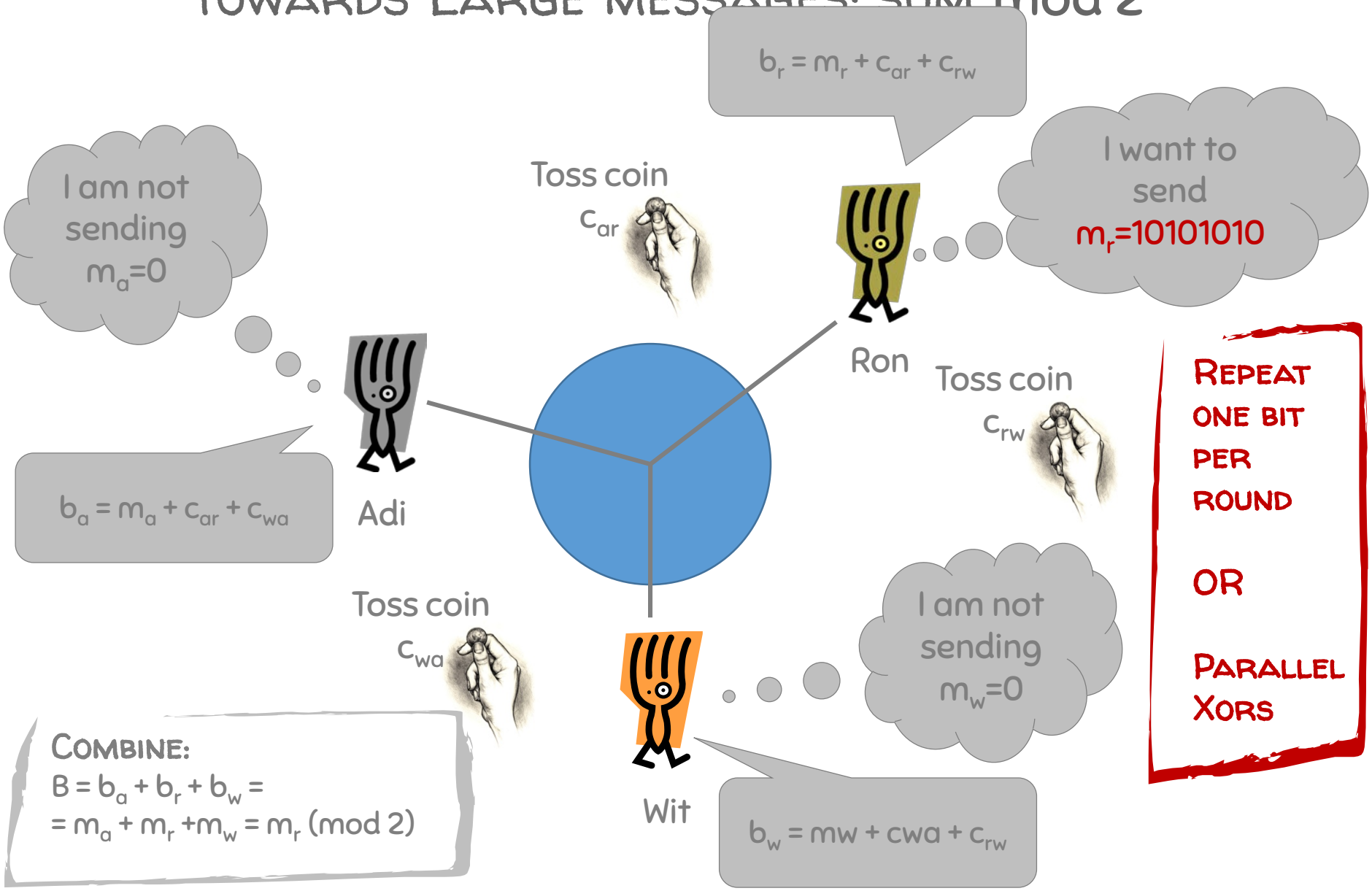
The Dining Cryptographers – Generalization

Towards Large messages: bit string



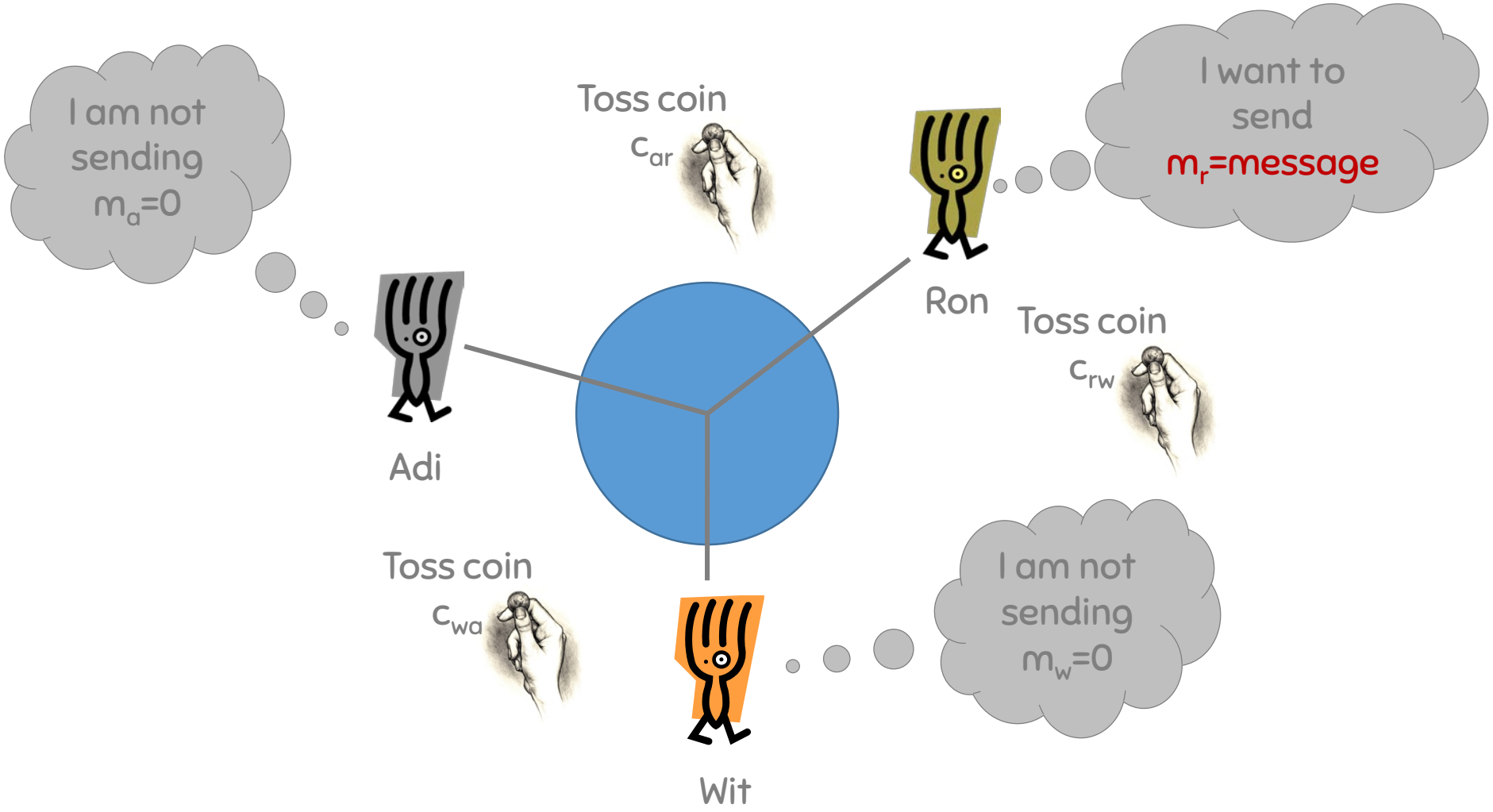
The Dining Cryptographers – Generalization

Towards Large messages: sum mod 2^m



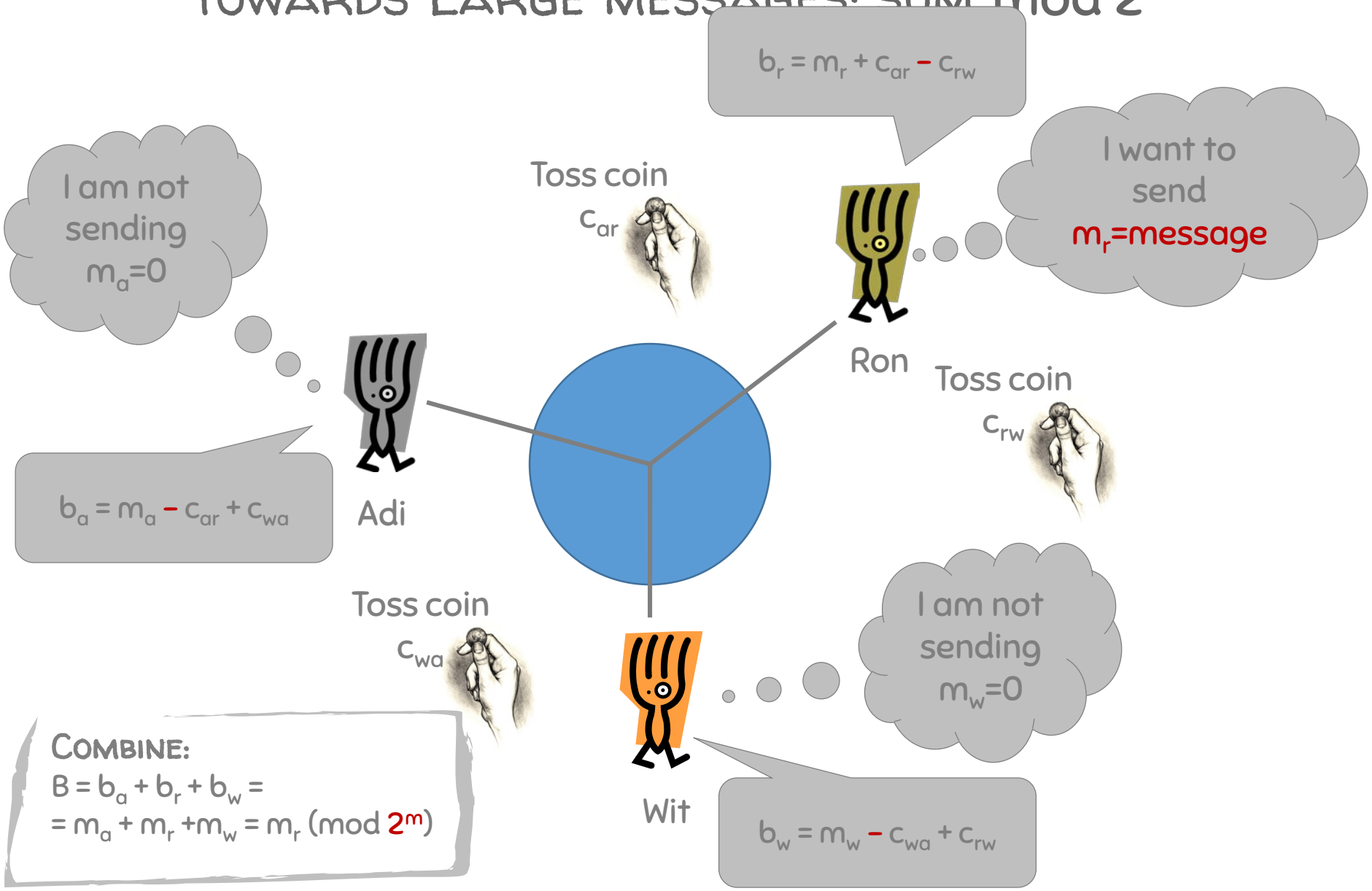
The Dining Cryptographers – Generalization

Towards Large messages: sum mod 2^m

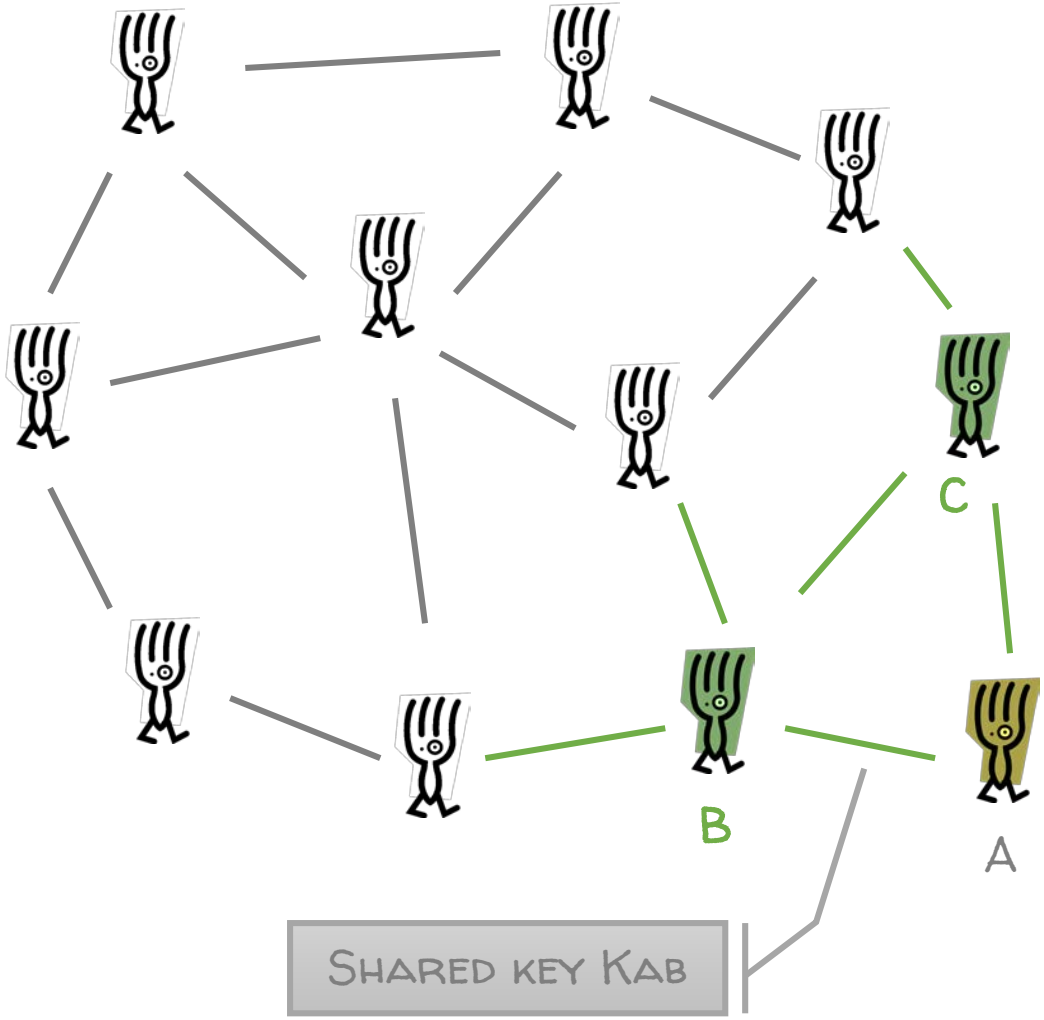


The Dining Cryptographers – Generalization

Towards Large messages: sum mod 2^m



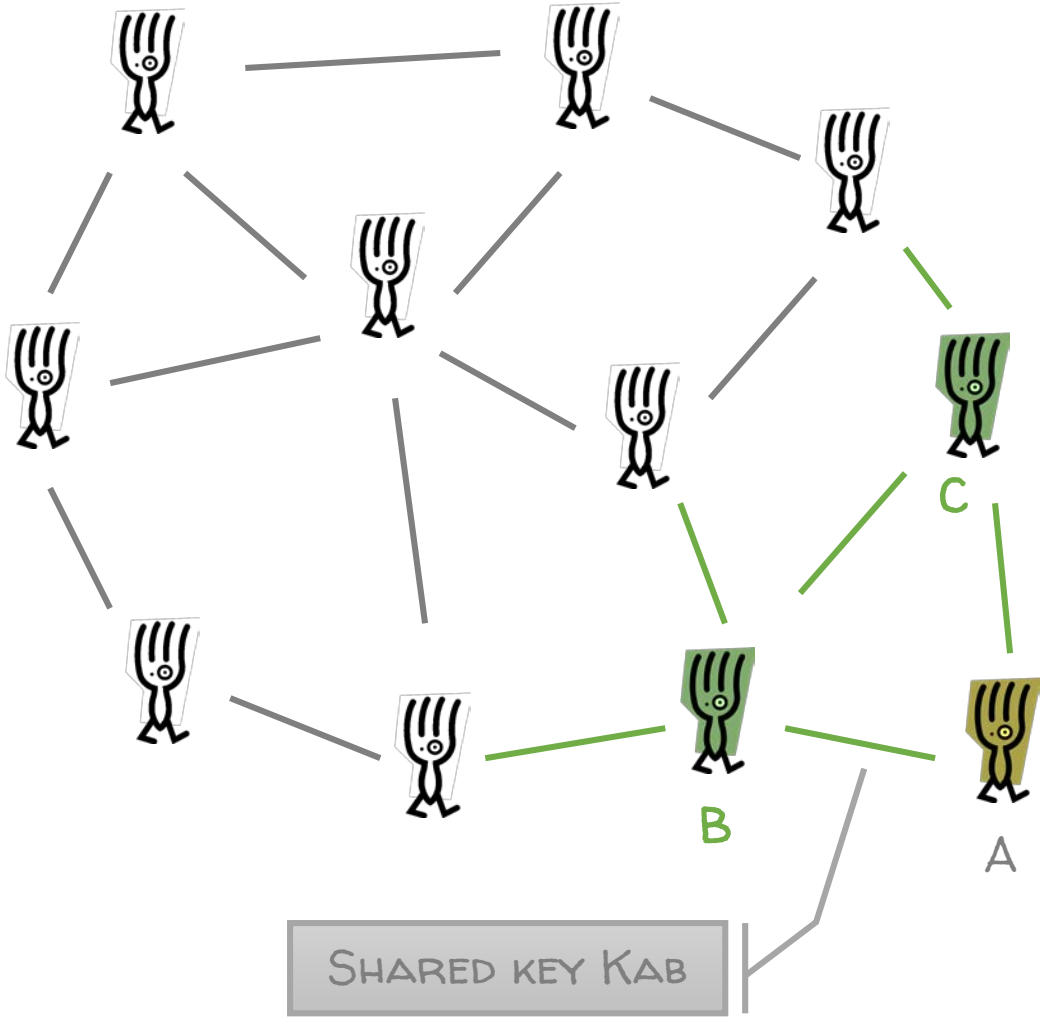
Key sharing graph – Security



Alice broadcasts

$b_a = c_{ab} + c_{ac} + m_a$

Key sharing graph – Security

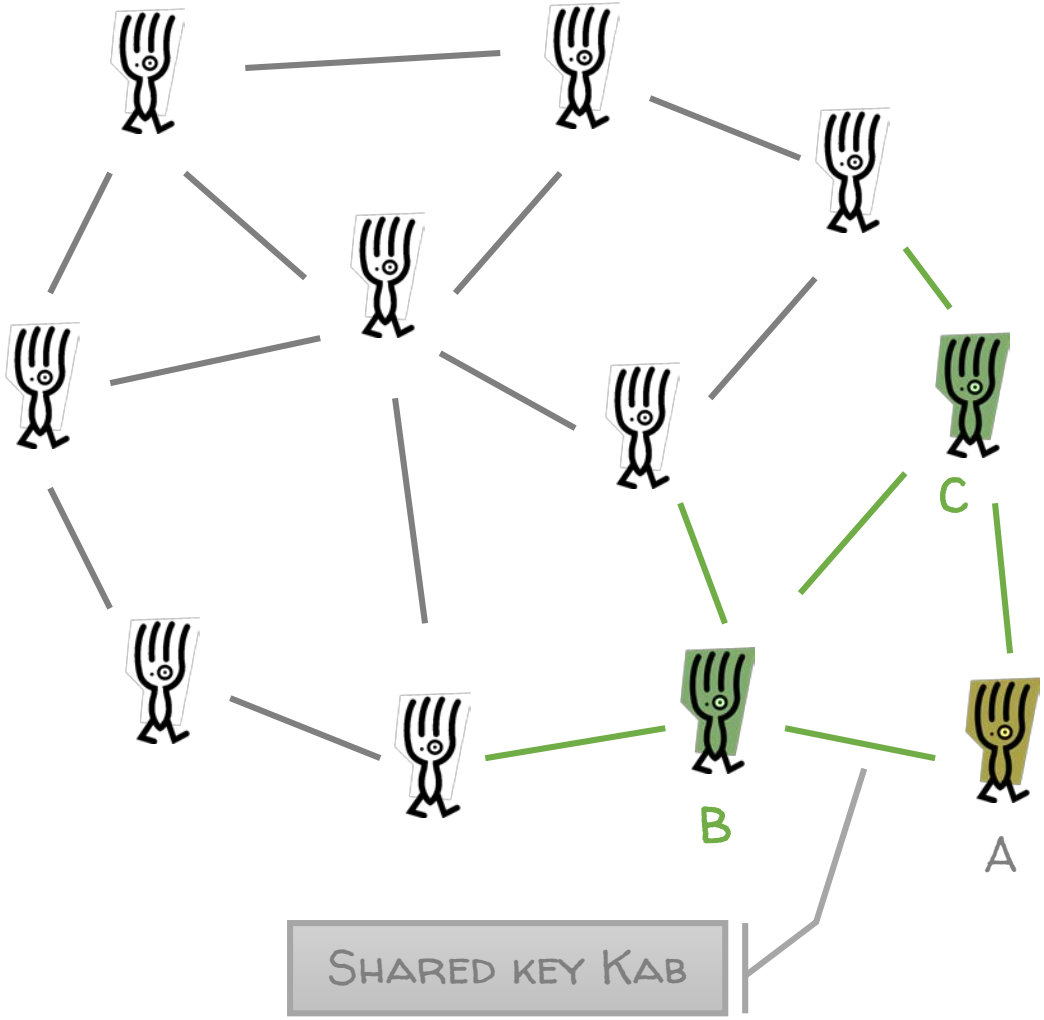


Alice broadcasts

$b_a = c_{ab} + c_{ac} + m_a$

If B and C corrupt

Key sharing graph – Security



Alice broadcasts

$b_a = c_{ab} + c_{ac} + m_a$

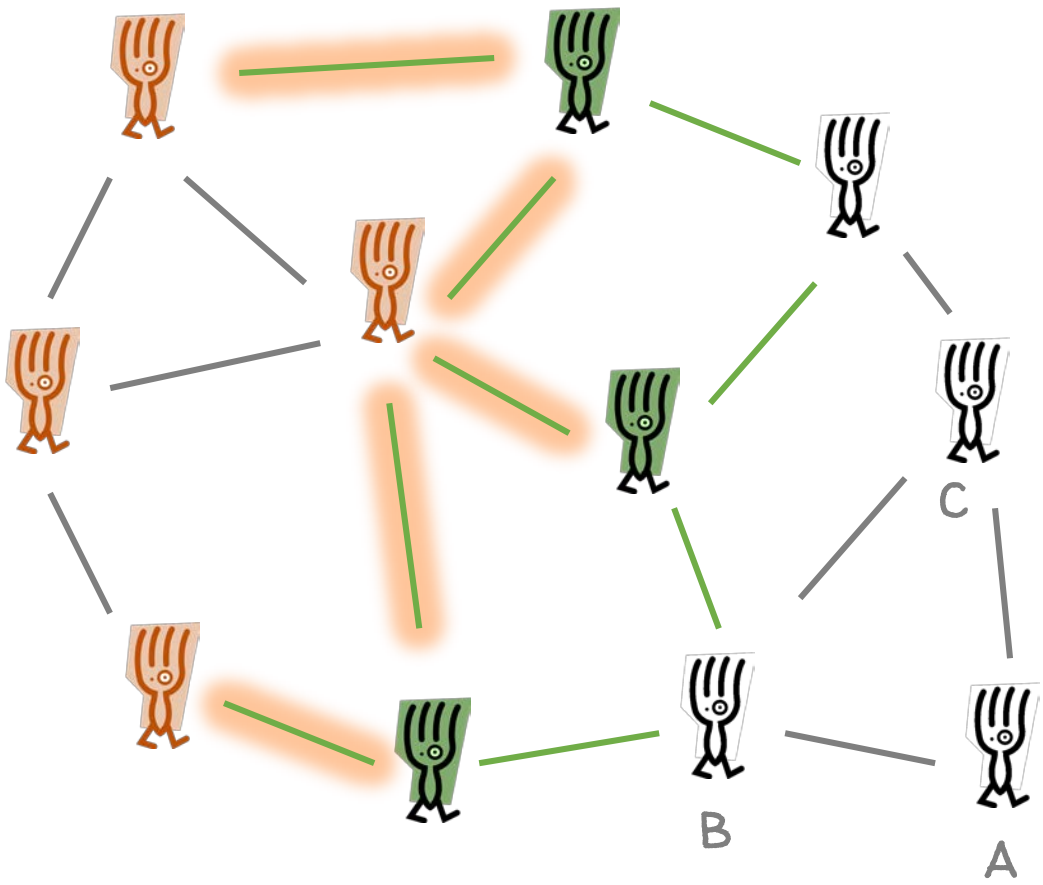
If B and C corrupt

Adversary's view

$b_a = c_{ab} + c_{ac} + m_a + c_{ab} + c_{ac}$

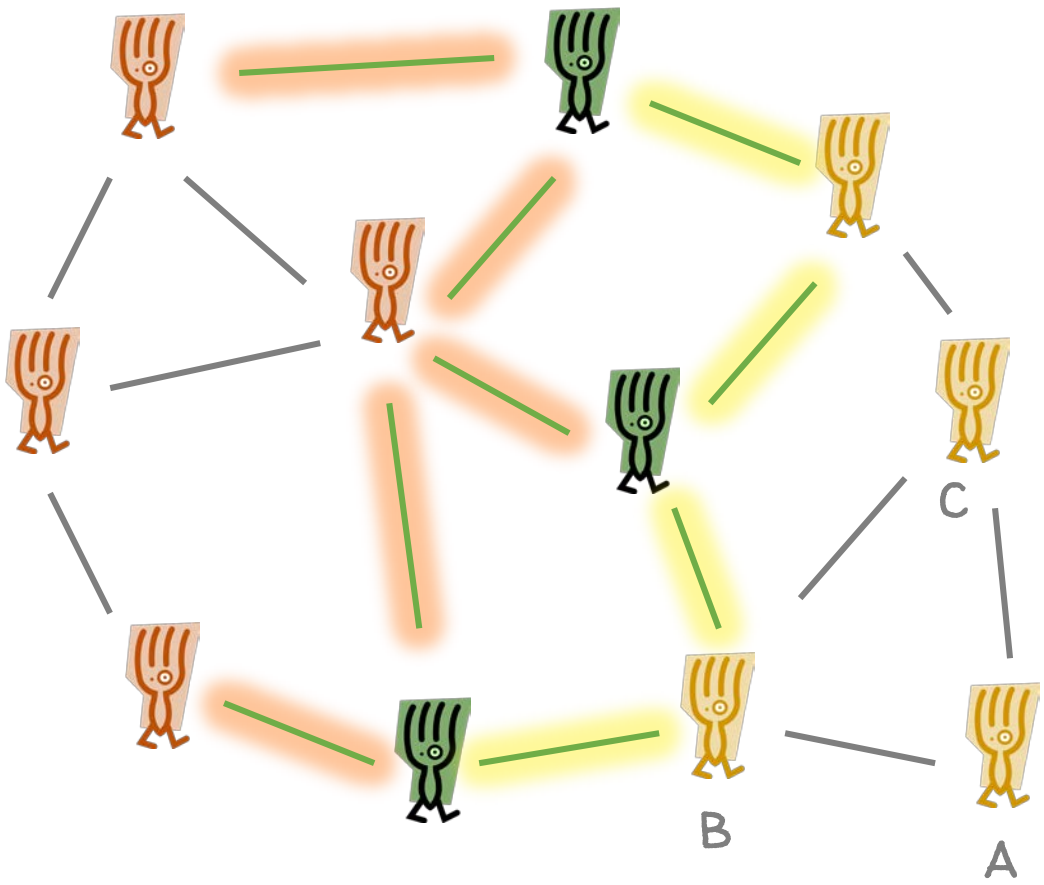
No Anonymity!!

Key sharing graph – Security



Adversary nodes partition the graph into a **red** and **yellow** sub-graphs

Key sharing graph – Security



Adversary nodes partition the graph into a **red** and **yellow** sub-graphs

Calculate:

$B_{red} = \sum b_j$, j is red

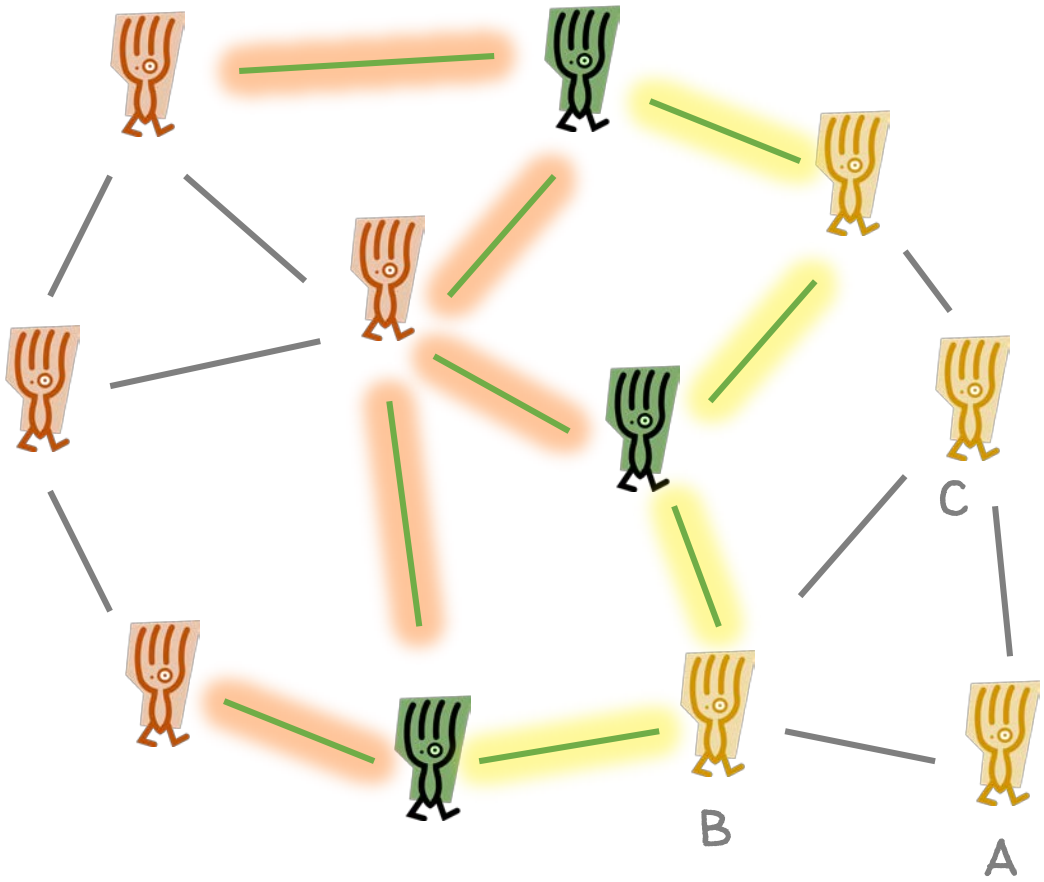
$B_{yellow} = \sum b_i$, i is yellow

Substract known keys

$B_{red} + K_{red-green} = \sum m_j$

$B_{yellow} + K'_{yellow-green} = \sum m_i$

Key sharing graph – Security



Adversary nodes partition the graph into a **red** and **yellow** sub-graphs

Calculate:

$B_{red} = \sum b_j$, j is red

$B_{yellow} = \sum b_i$, i is yellow

Substract known keys

$B_{red} + K_{red-green} = \sum m_j$

$B_{yellow} + K'_{yellow-green} = \sum m_i$

Discover the originating subgraph

Reduction in anonymity!!

Anonymity set size = 4 (not 11 or 8!)

Implementing DC-nets

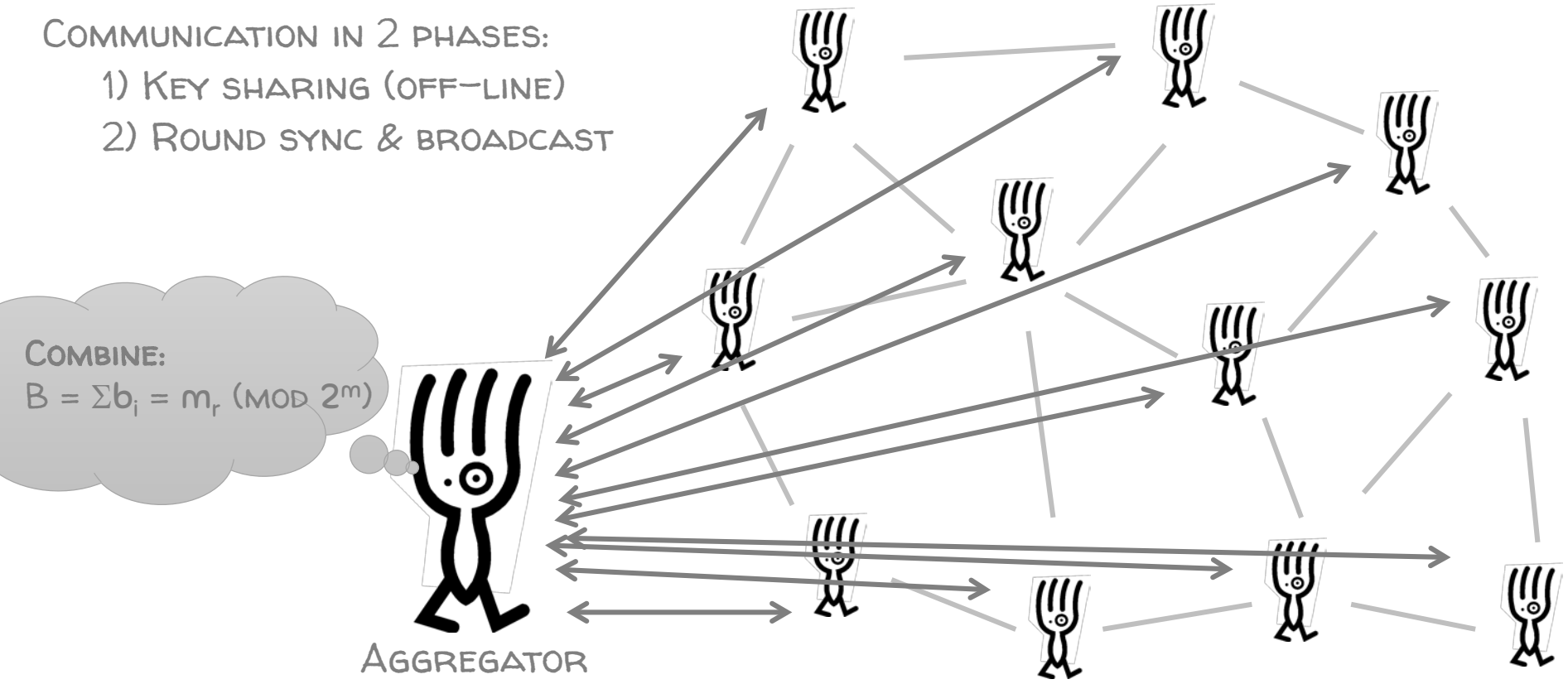
b_i broadcast graph

No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast



Implementing DC-nets: P2P

b_i broadcast graph

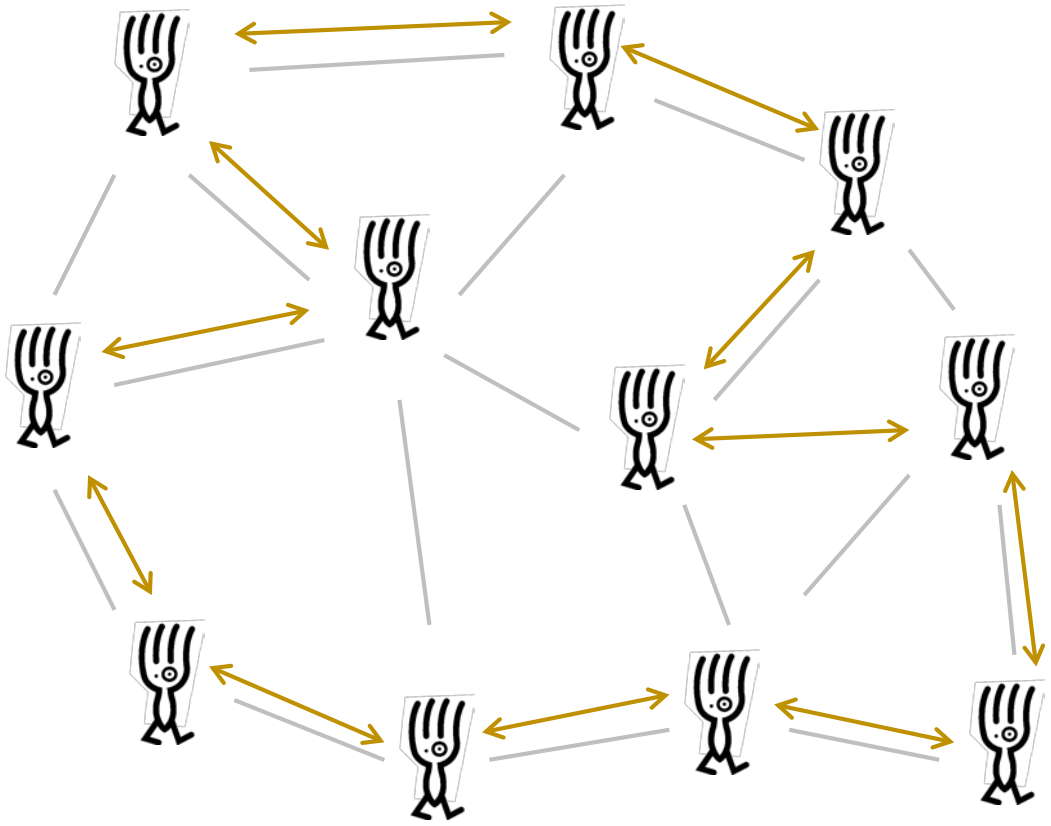
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Implementing DC-nets: P2P

b_i broadcast graph

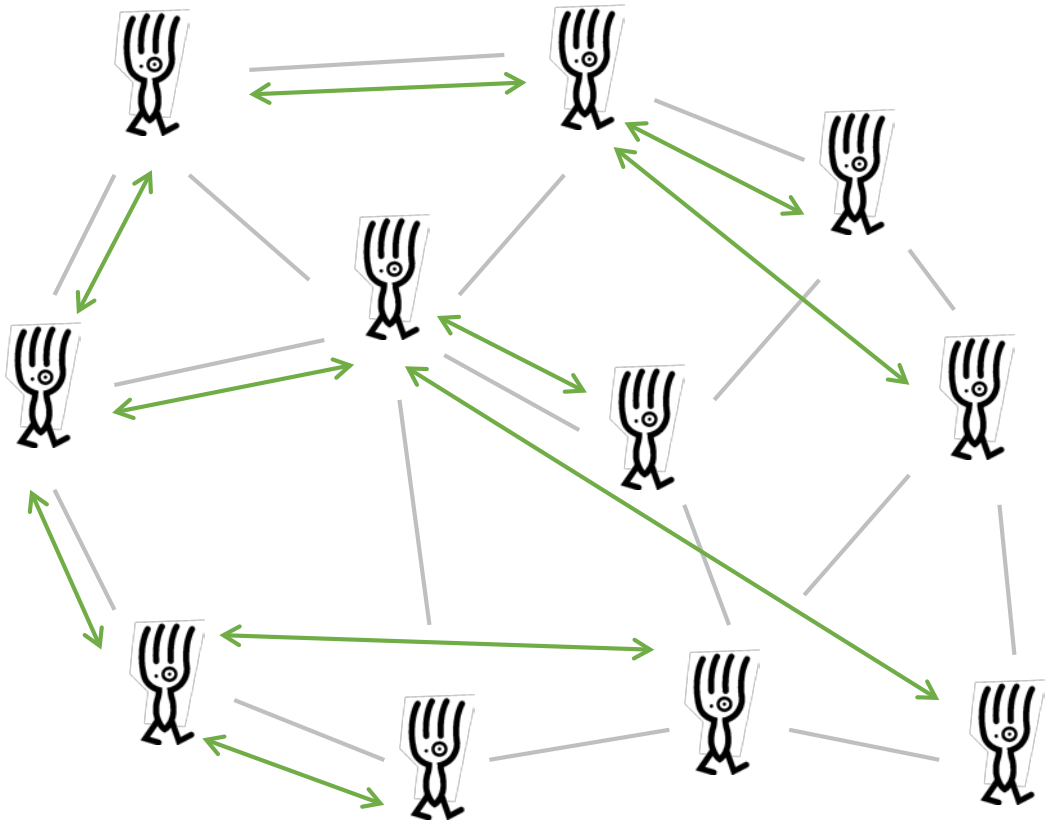
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

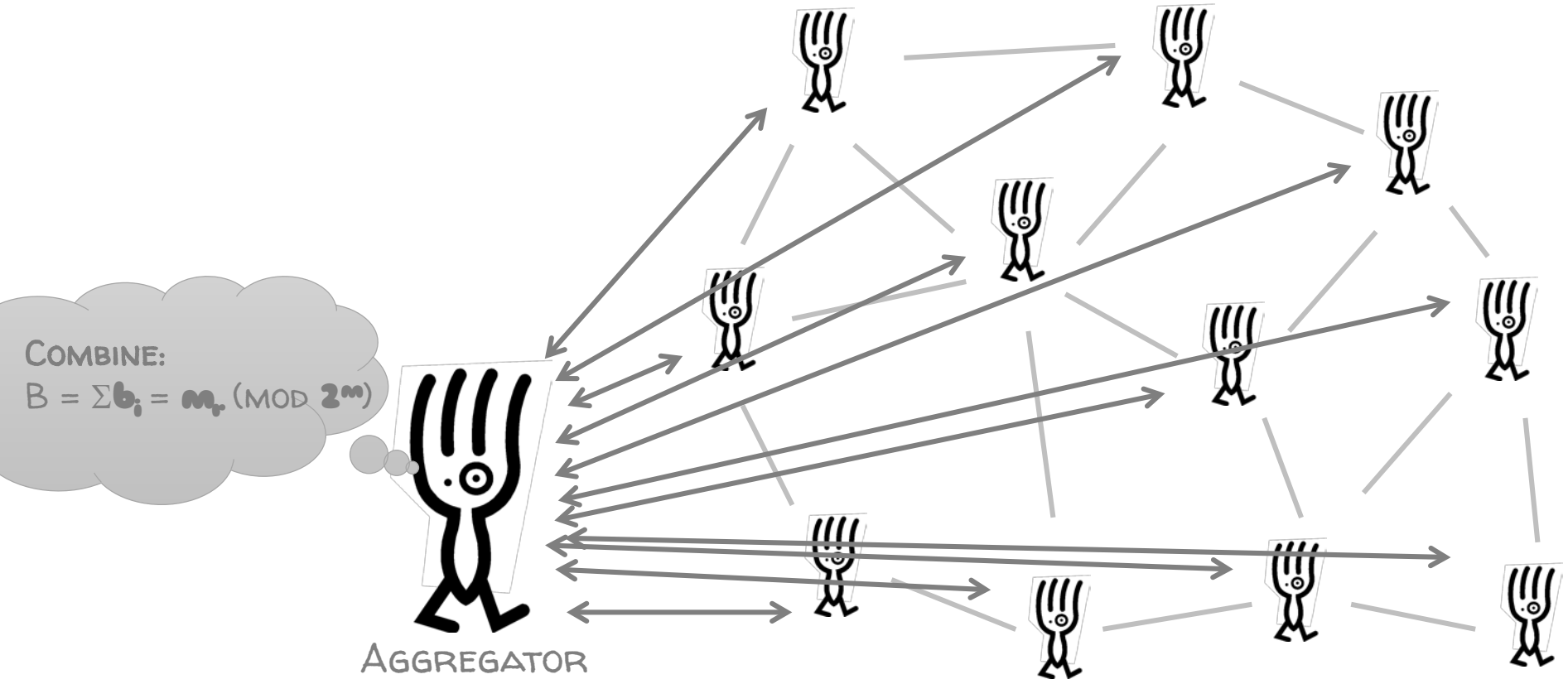
Tree?



Predecessor attack, does it work?

NO!!!

b_i broadcast graph



Predecessor attack, does it work?

b_i broadcast graph

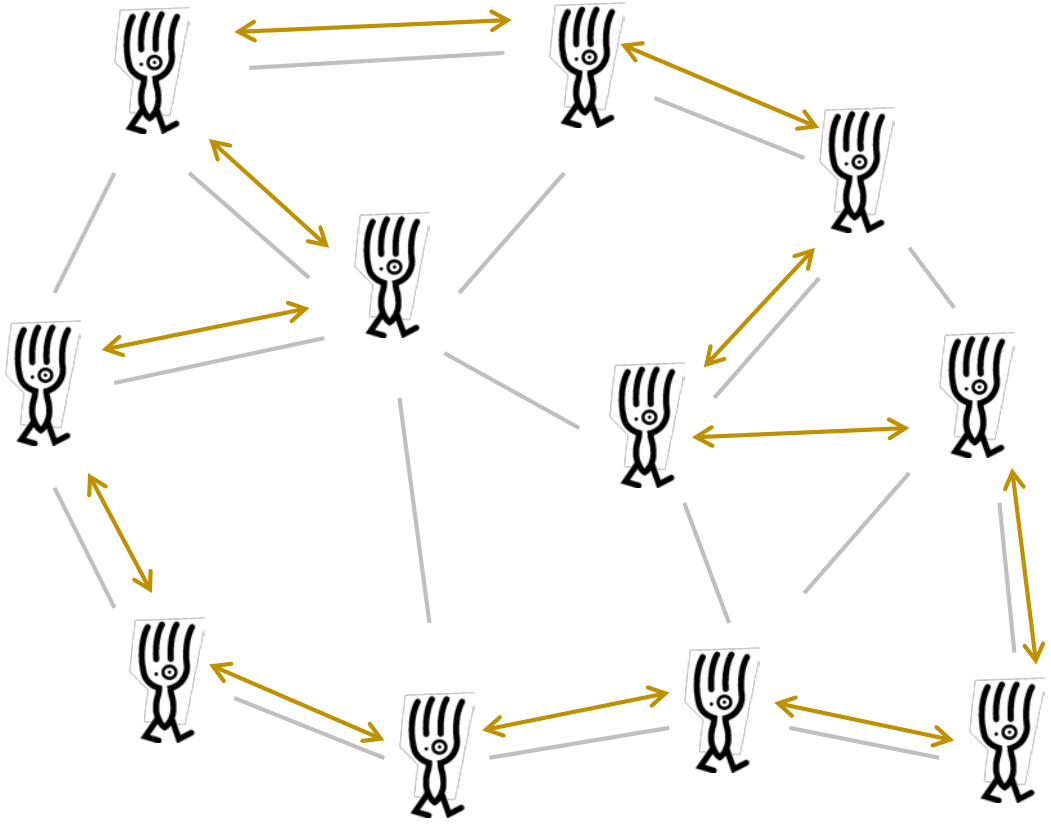
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

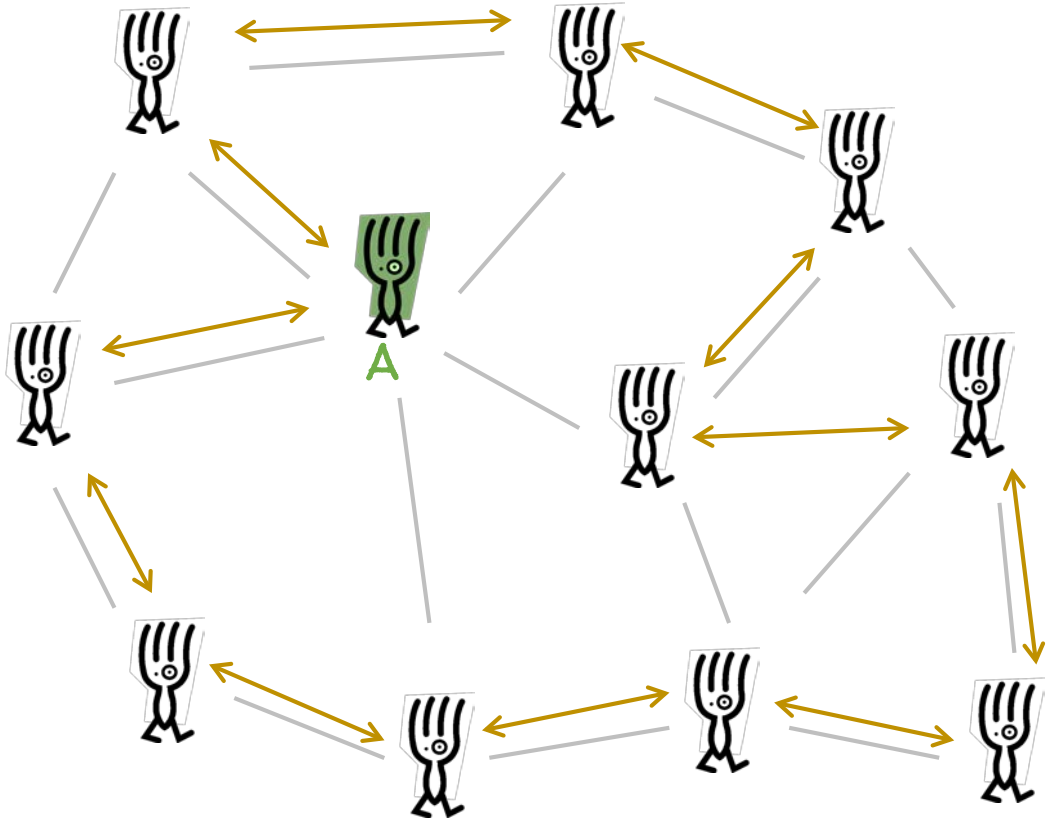
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

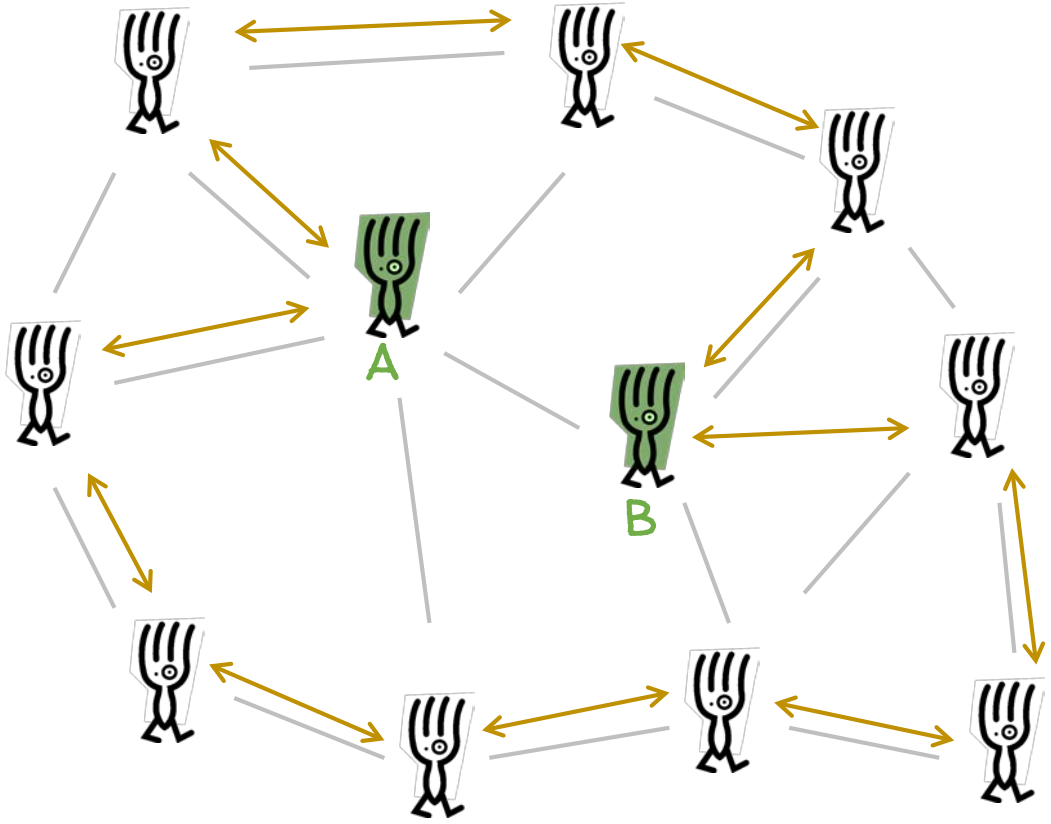
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

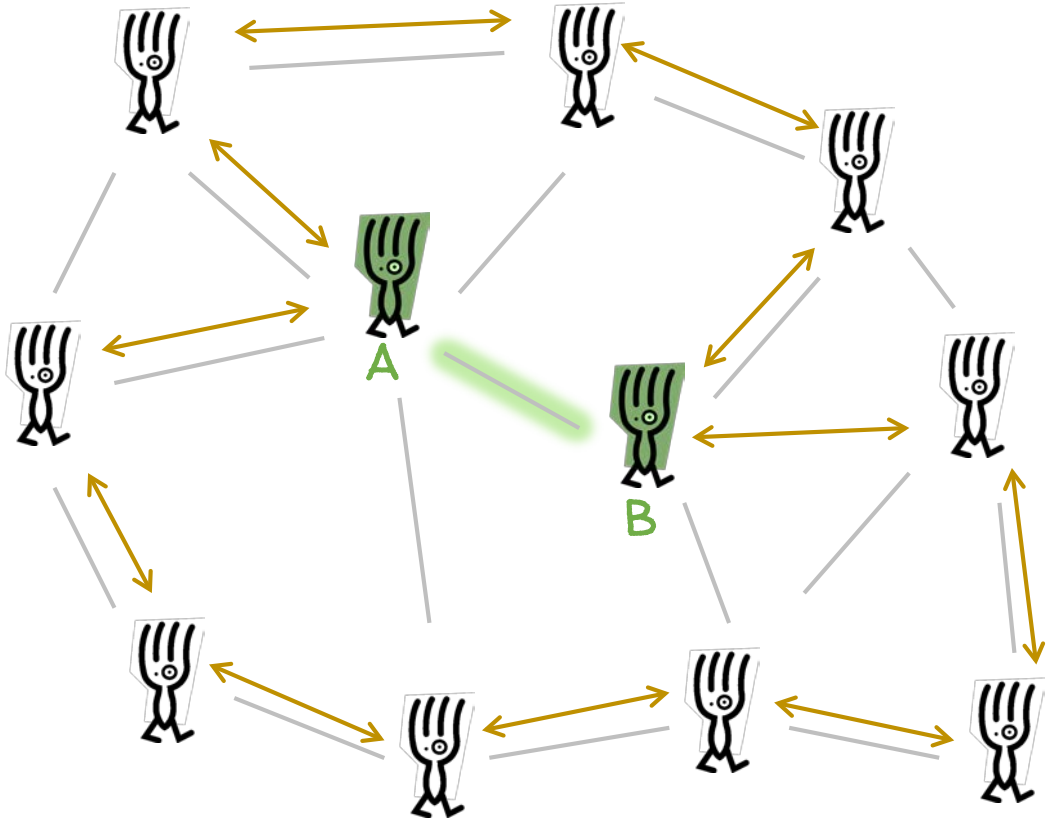
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

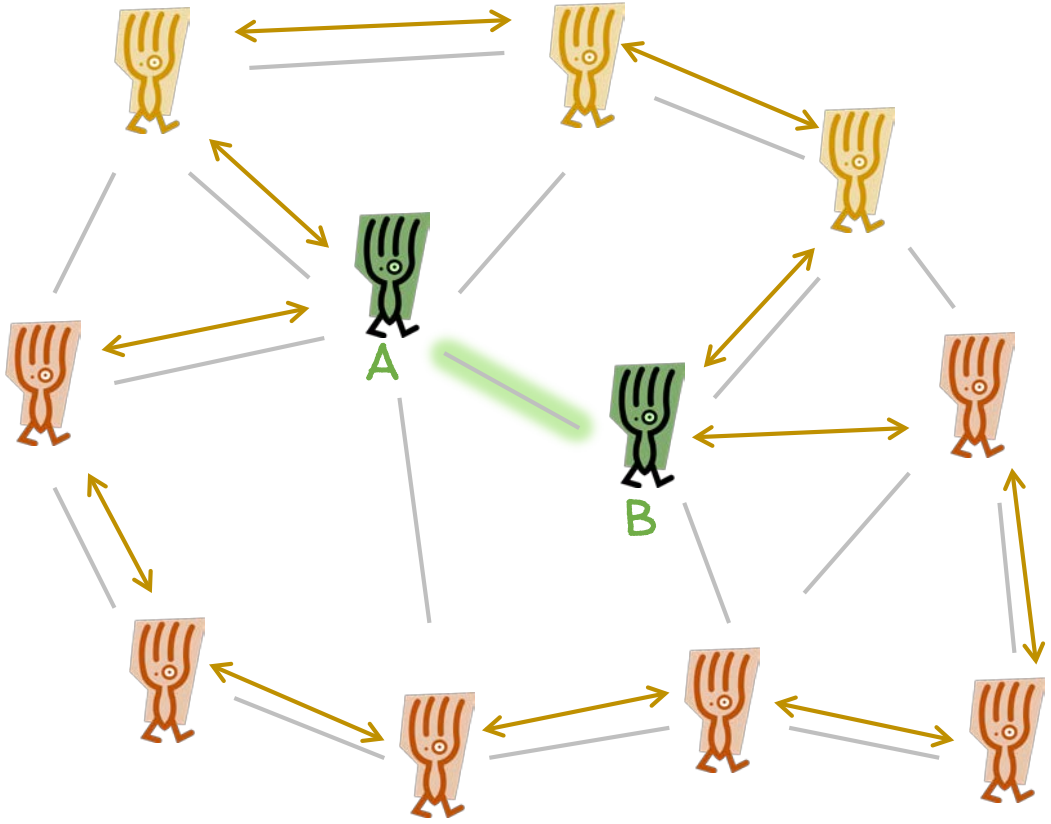
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

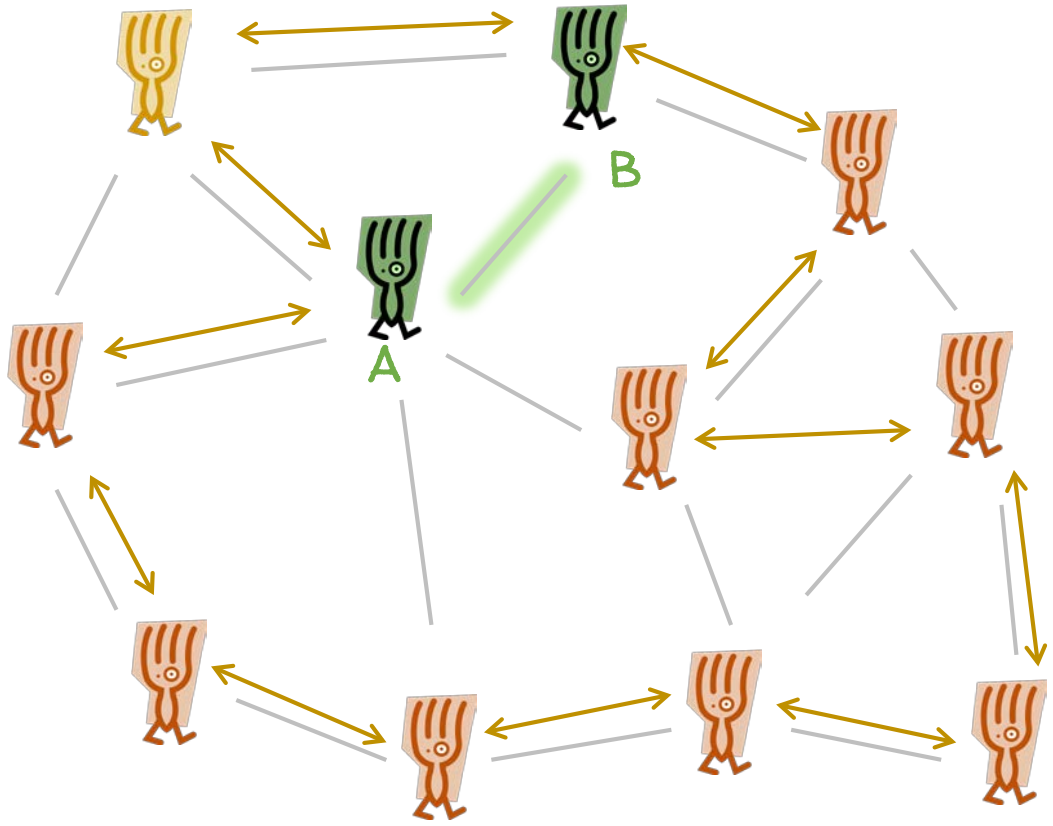
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

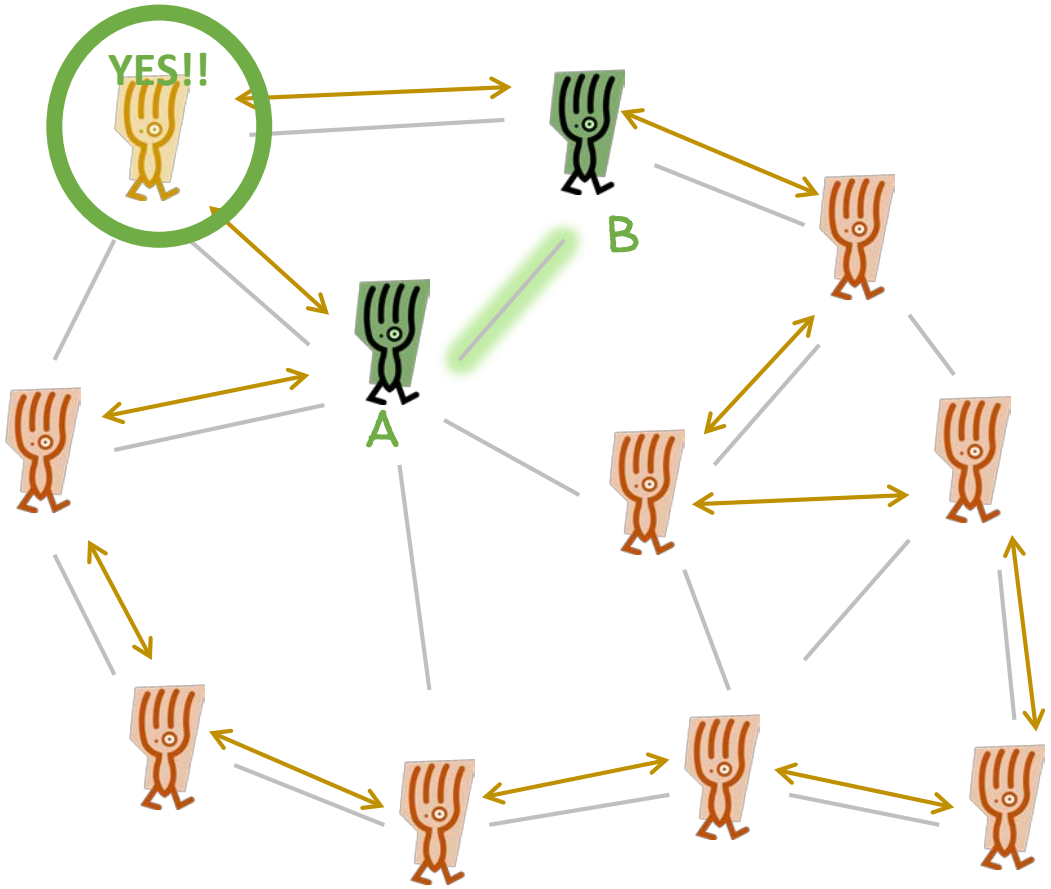
No DoS unless split in graph

Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?



Predecessor attack, does it work?

b_i broadcast graph

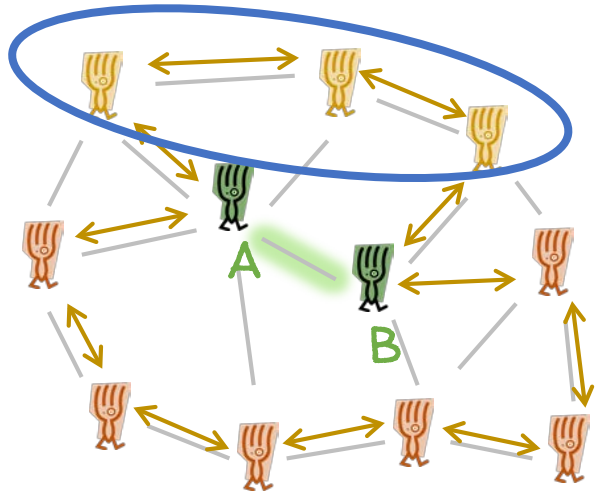
No DoS unless split in graph

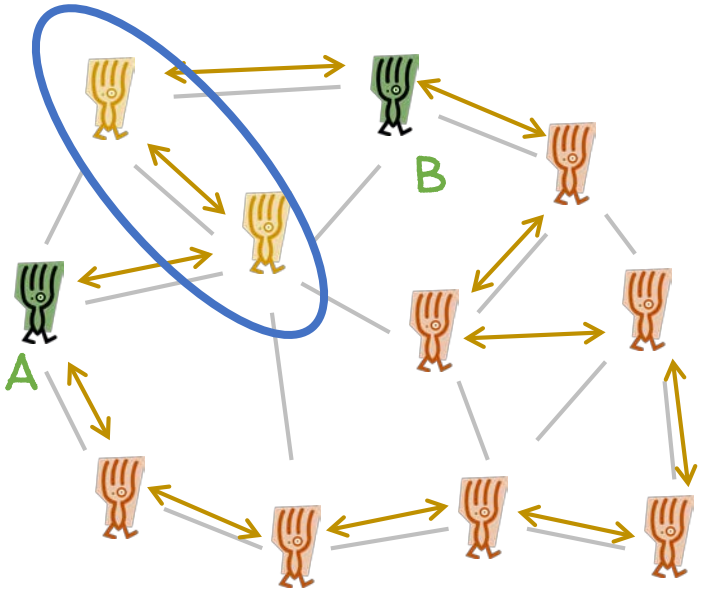
Communication in 2 phases:

1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

Ring?





Predecessor attack, does it work?

b_i broadcast graph

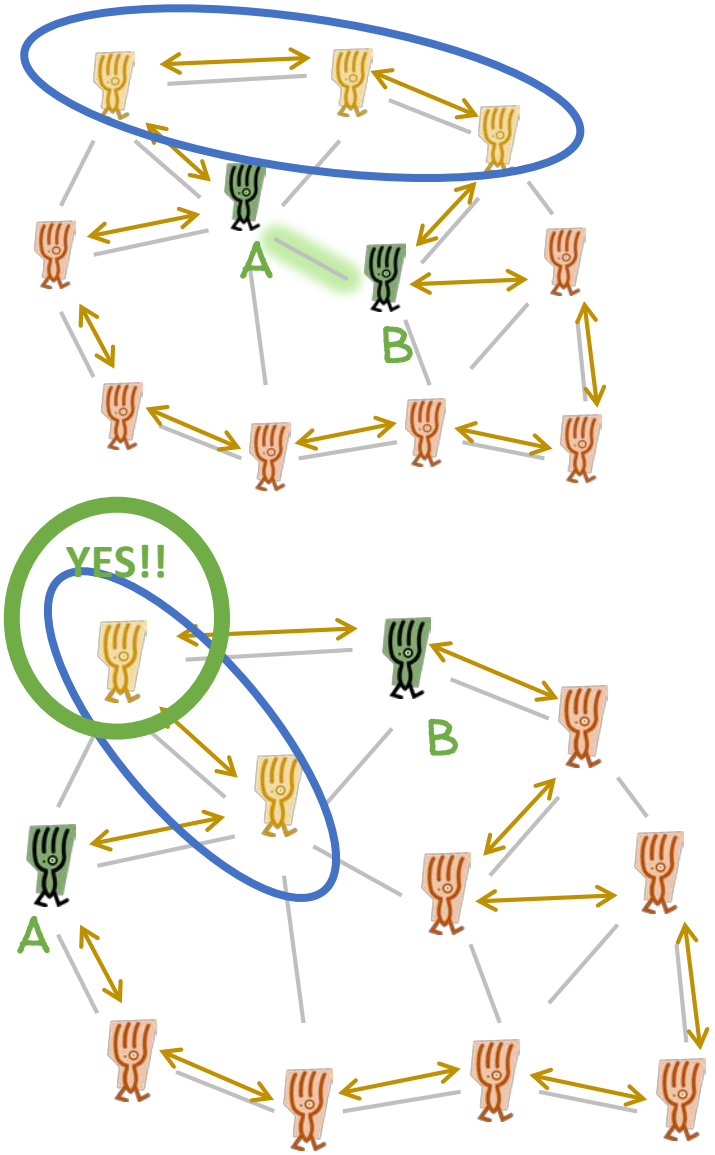
No DoS unless split in graph

Communication in 2 phases:

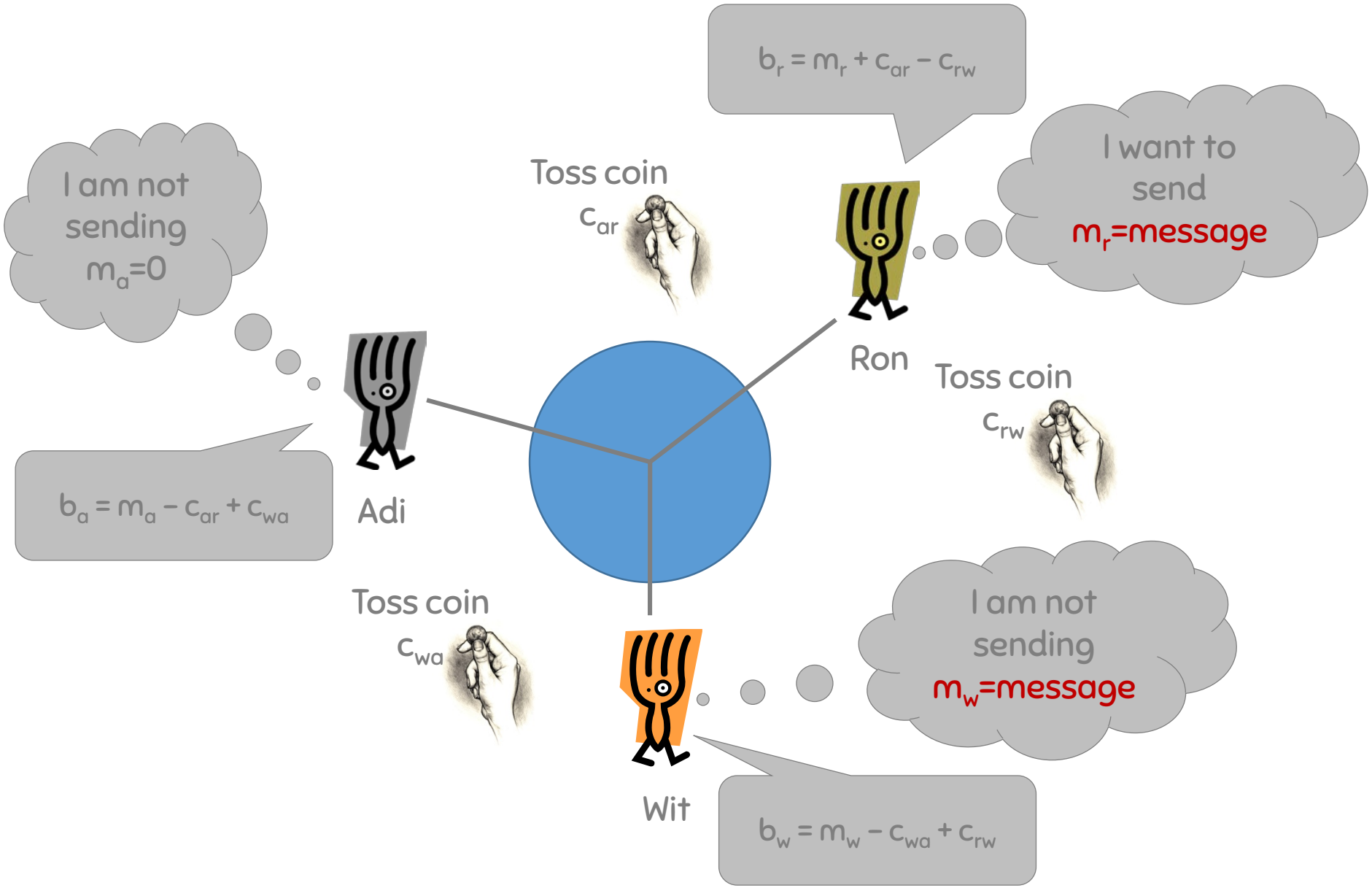
1) Key sharing (off-line)

2) Round sync & broadcast (peer-to-peer?)

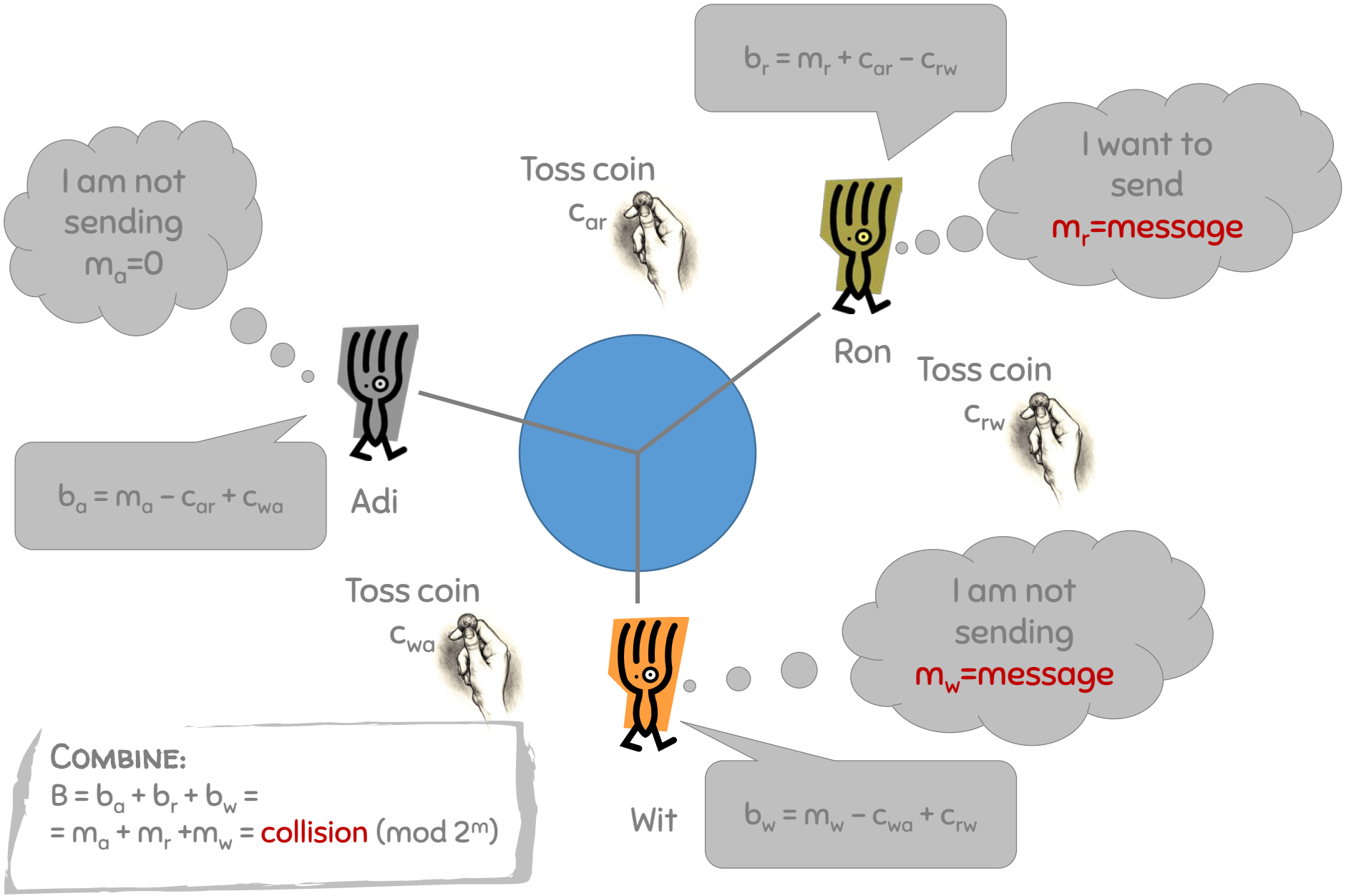
Ring?



The Dining Cryptographers – Collisions



The Dining Cryptographers – Collisions



How to resolve collisions?

Ethernet: detect collision and random re-transmission

DC-nets: Collisions do not destroy all information

$B = b_a + b_r + b_w = m_a + m_r + m_w =$
$= \text{collision} \pmod m$
$= \text{message}_1 + \text{message}_2 \pmod m$

How to resolve collisions?

Ethernet: detect collision and random re-transmission

DC-nets: Collisions do not destroy all information

$B = b_a + b_r + b_w = m_a + m_r + m_w =$
$= \text{collision} \pmod m$
$= \text{message}_1 + \text{message}_2 \pmod m$

N collisions can be decoded in N transmissions!

DC-net takeaways

- Security is great!
  - Full key sharing graph ⇔ perfect anonymity
- Communication cost – BAD
  - (N broadcasts for each message!)
  - Naive: $O(N^2)$ cost, O(1) Latency
  - Not so naive: O(N) messages, O(N) latency
    - Ring structure for broadcast
  - Expander graph: O(N) messages, O(logN) latency?
  - Centralized: O(N) messages, O(1) latency
- Not practical for large(r) N! ☹
  - Local wireless communications?
- Perfect Anonymity

Hervibore

Entry control

- Distribute nodes
- Avoid choice
- Cost to enter

Round

- Reserve
- Transmission
- Exit
  - (avoid intersection)

min(size)=k

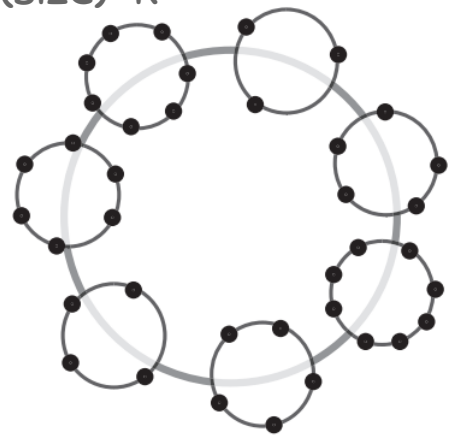


Fig. 1. The global topology of Herbivore is structured as a ring, with each clique assigned a unique key. While communication in Herbivore is primarily done at the clique level, nodes can leverage this global backbone to communicate with other cliques

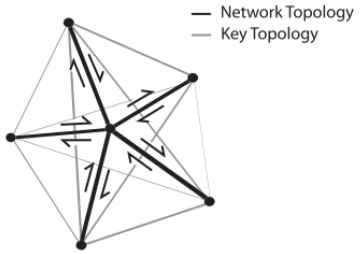


Fig. 3. A Six Node Clique. While the network topology is star-shaped to ensure high bandwidth utilization and low latency, the key topology is a complete graph to protect anonymity. Since the center of the star has a disproportionate network load, Herbivore selects a new center for each round by cycling through the clique members.

We have seen several techniques for anonymous communications

And different attacks

Next week

Traffic Analysis:
Protocols, Attacks, Design Issues, and Open Problems.