# MPC for MPC: Secure Computation on a Massively Parallel Computing Architecture

Hubert Chan, Kai-Min Chung, **Wei-Kai Lin**, Elaine Shi

2019/01/13

# Models of Parallel Computation

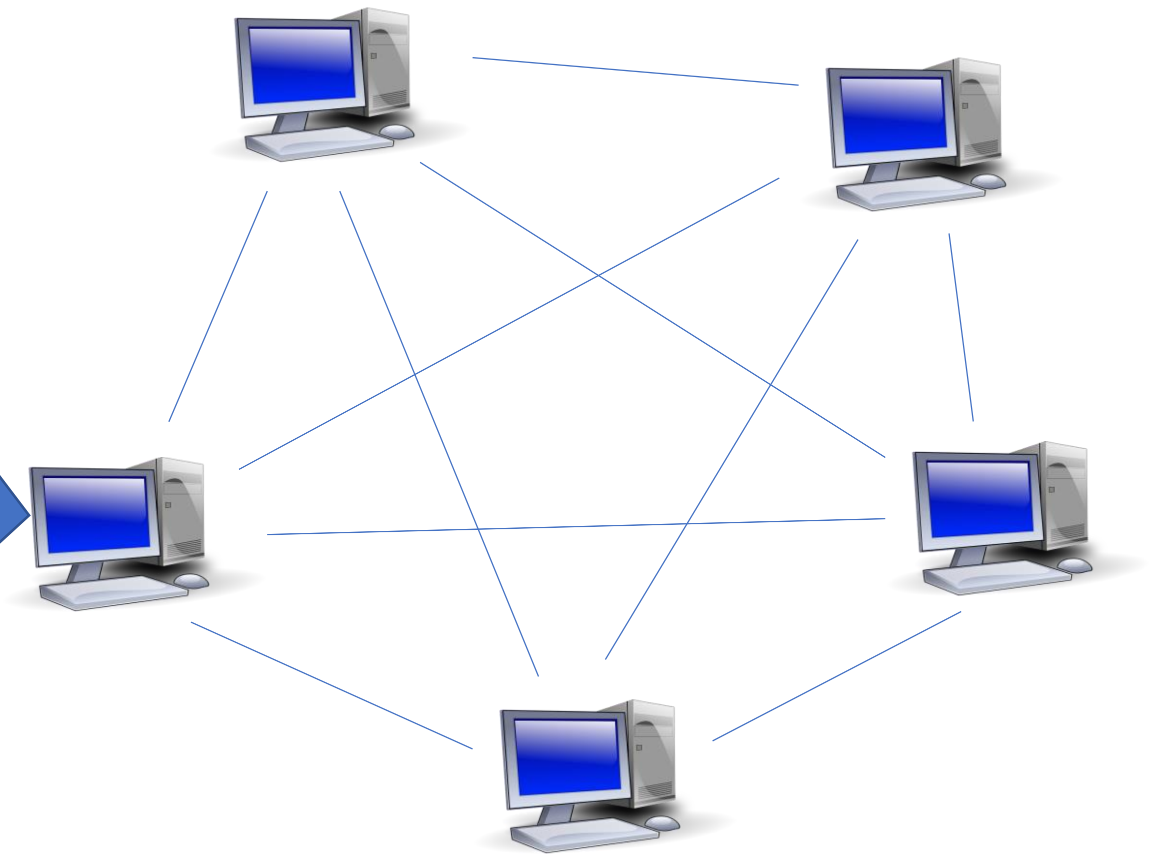
- Circuit?
- Parallel Random Access Machine (PRAM)
- Bulk Synchronous Parallel (BSP) model

Karloff, Suri, and Vassilvitskii (SODA 2010)

**Massively Parallel Computation, MPC**

## Massively Parallel Computation (MPC)

- $m$ Random Access Machines (RAM)
- Fully connected
- Each of space $s$



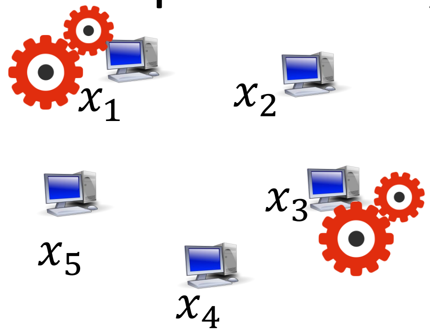
- Input size $N$
- $s = N^{\epsilon}$, const $\epsilon \in (0,1)$
- $\Rightarrow m \geq N^{1-\epsilon}$

# MPC Proceeds in Rounds

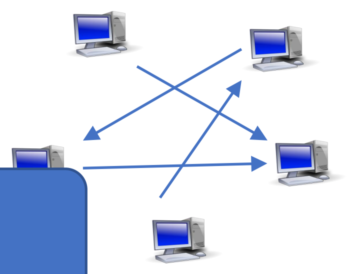
Space $s = N^{\epsilon}$

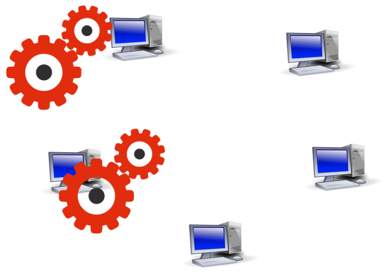
Compute locally

Send messages (send/receive $s$ bits each)

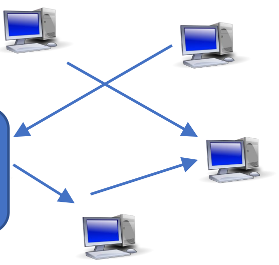


Round 1





Round 2

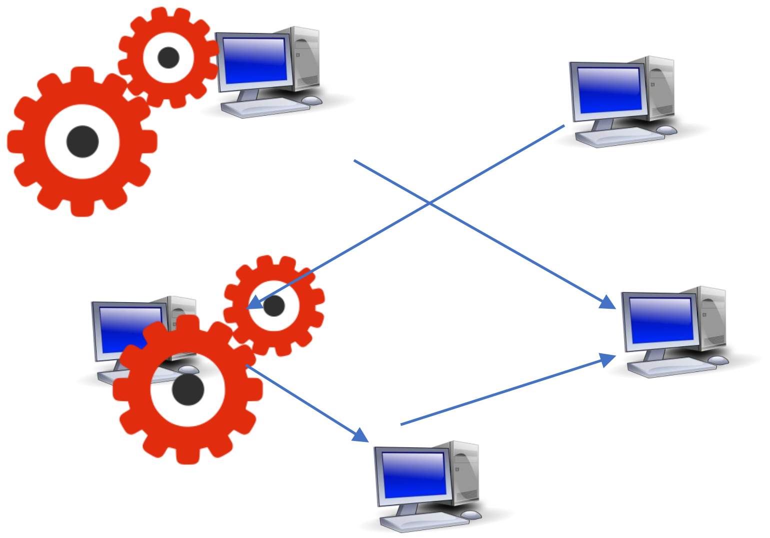


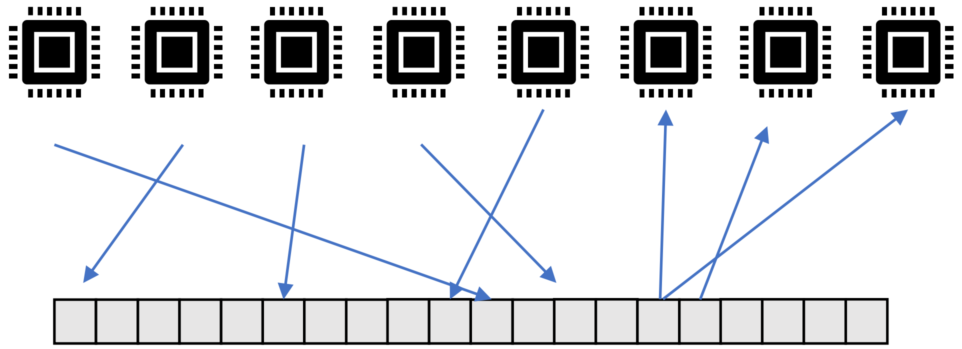
Repeat rounds .... Output jointly

Major Metric: Round Complexity

MPC
Space $s = N^{\epsilon}$

PRAM





# Rounds

# Parallel steps

Sort $N$ items

$O(1)$

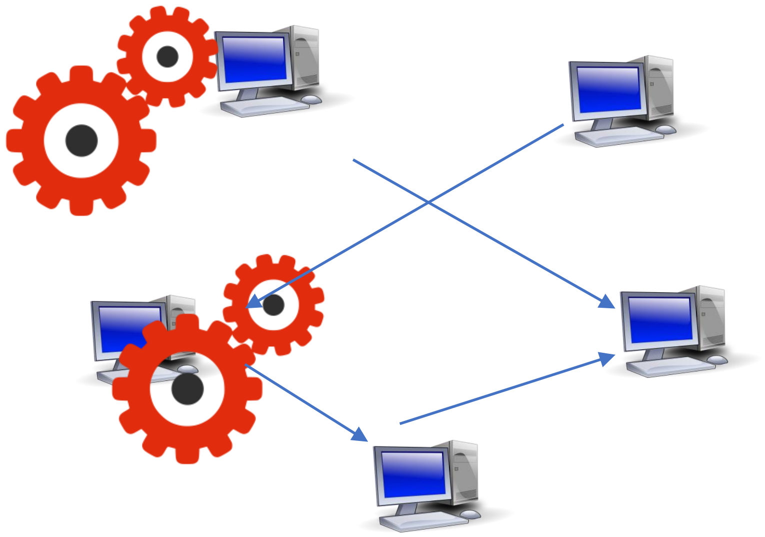
$\Omega(\log N)$

Same reason motivated MPC (than PRAM) also motivated that …

## Question:
## How to get MPC algo “secure”?
## What is the cost?

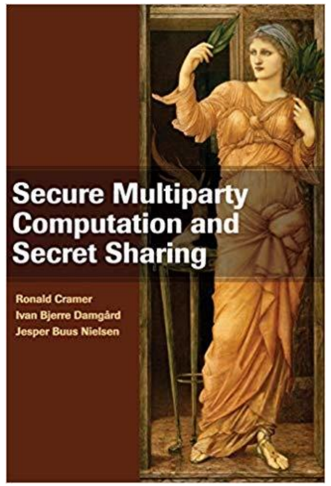
✓ MPC

Space $s = N^{\epsilon}$



What is “secure” in MPC model?

Many adversarial settings …

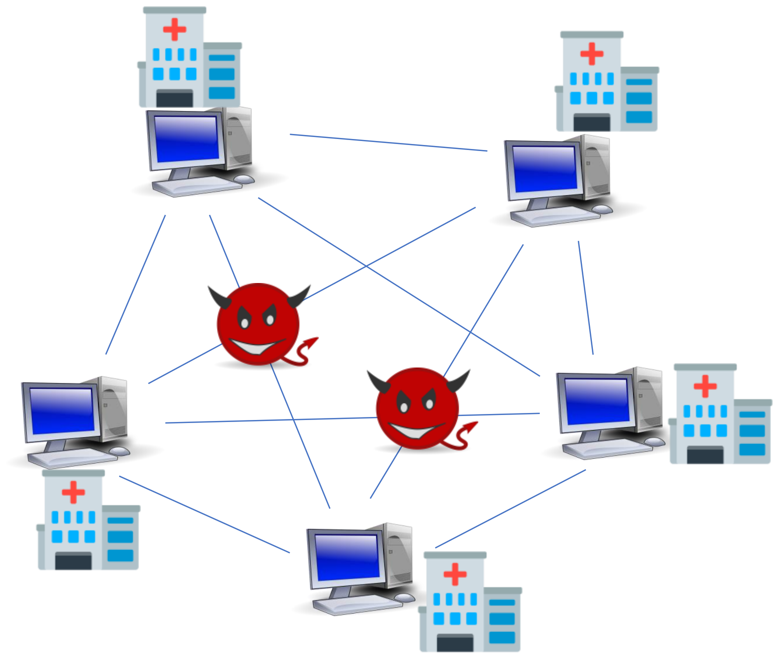


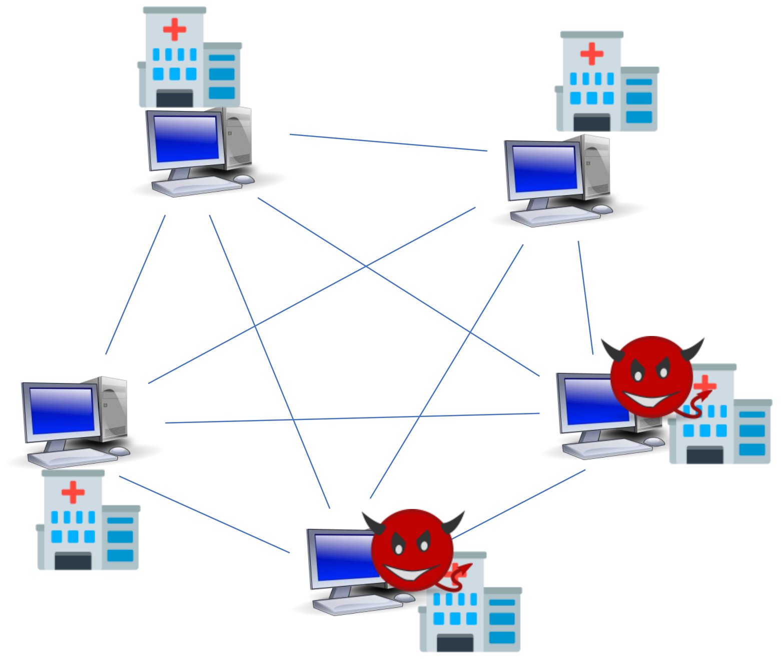
~~In cryptography, MPC =~~

~~secure Multiparty Computation~~

Scenario 1:
Adversary is only eavesdropping, wants to learn secret input

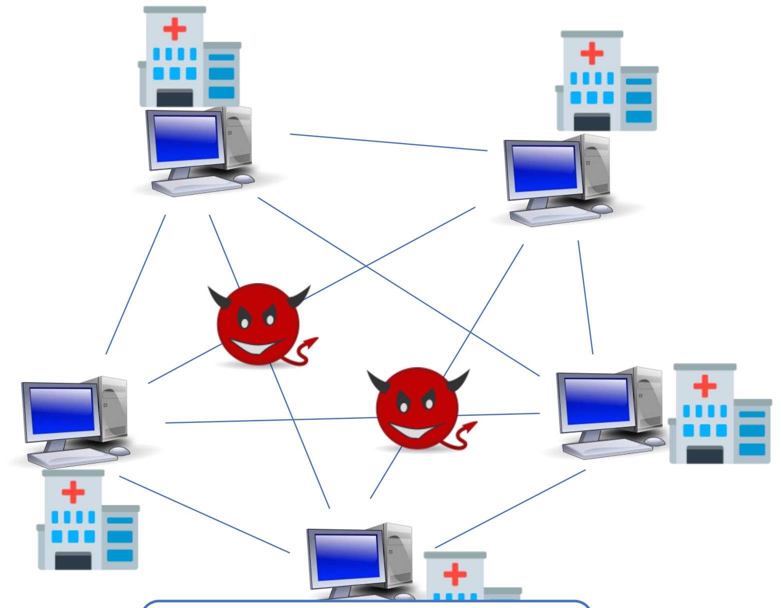
Scenario 2:
Adversary corrupts some machines, wants secret on others





# Scenario 1: **Constant Overhead**

Adversary is only eavesdropping, wants to learn secret input



Space $s = N^{\epsilon}$

MPC algo taking space $s$, rounds $R$

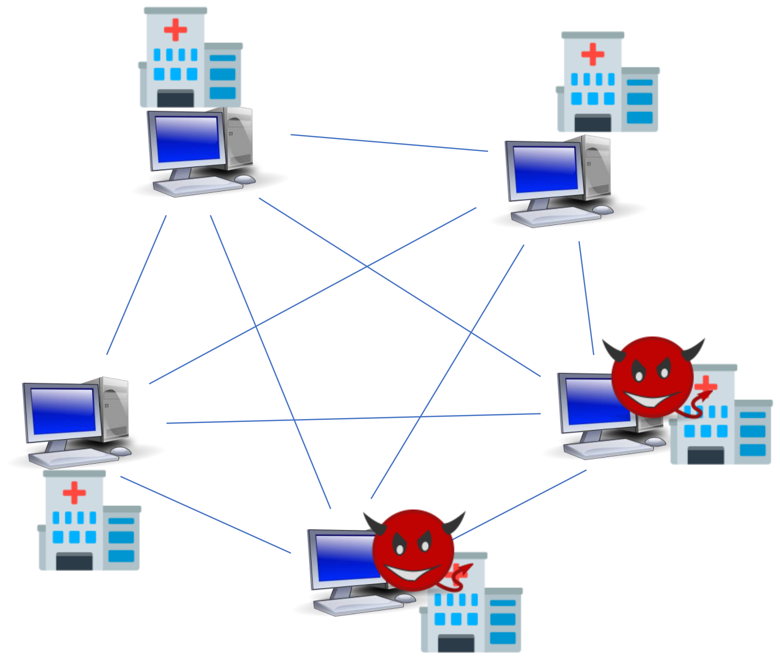
$\Downarrow$

secure MPC algo taking space $O(s)$, rounds $O(R)$

Failure probability in correctness: $\exp\left(-\Omega(\sqrt{s})\right)$

# Scenario 2: **Constant in Rounds, Security-Parameter in Space**

Adversary corrupts some <u>machines</u>, wants secret on <u>others</u>



MPC algo taking space $s$, rounds $R$

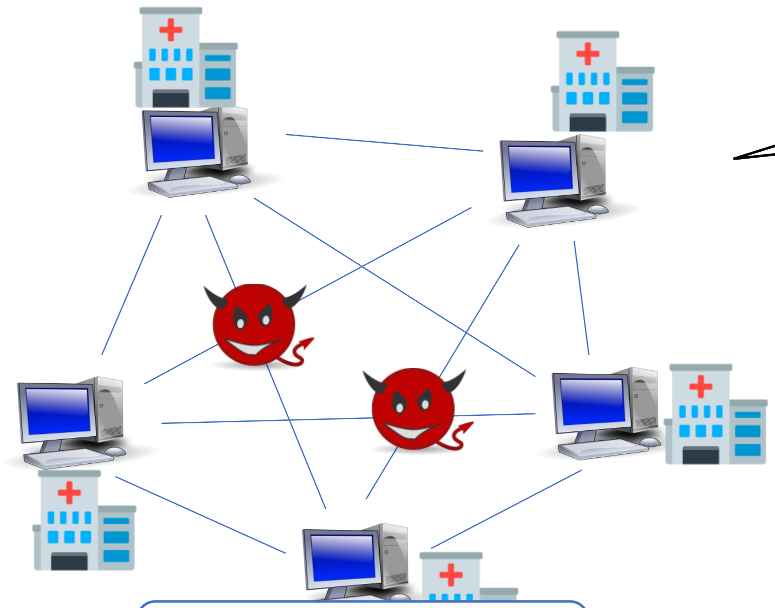
$\Downarrow$

secure MPC algo taking space $O\left(s \cdot \underline{poly(\kappa)}\right)$, rounds $O(R)$

Assume Learning With Errors (LWE), compact Fully Homomorphic Encryption (FHE), and corrupt machines < 1/3.

Fail probability in correctness: $\exp\left(-\Omega(\sqrt{s})\right)$.

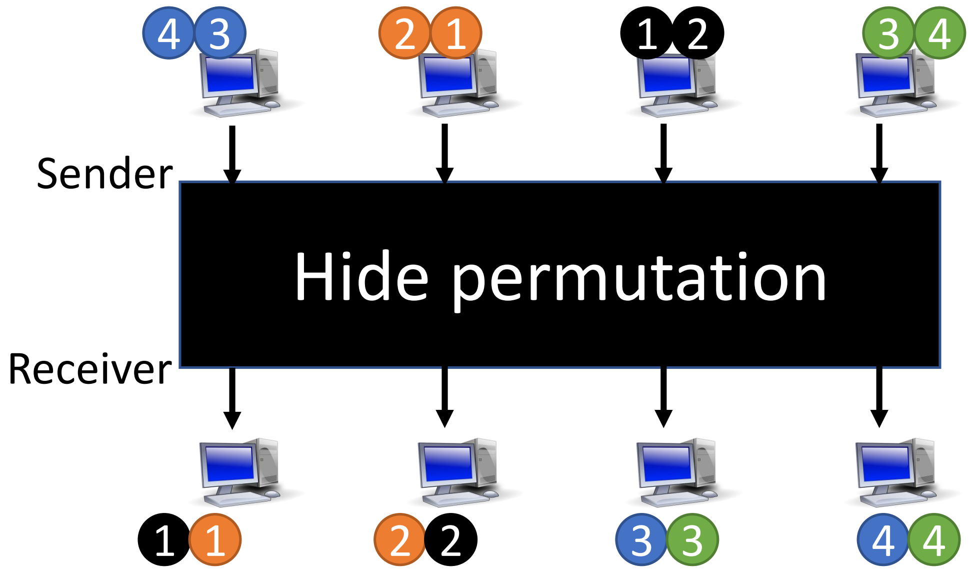
## (Scenario 1) Technique: Oblivious Routing



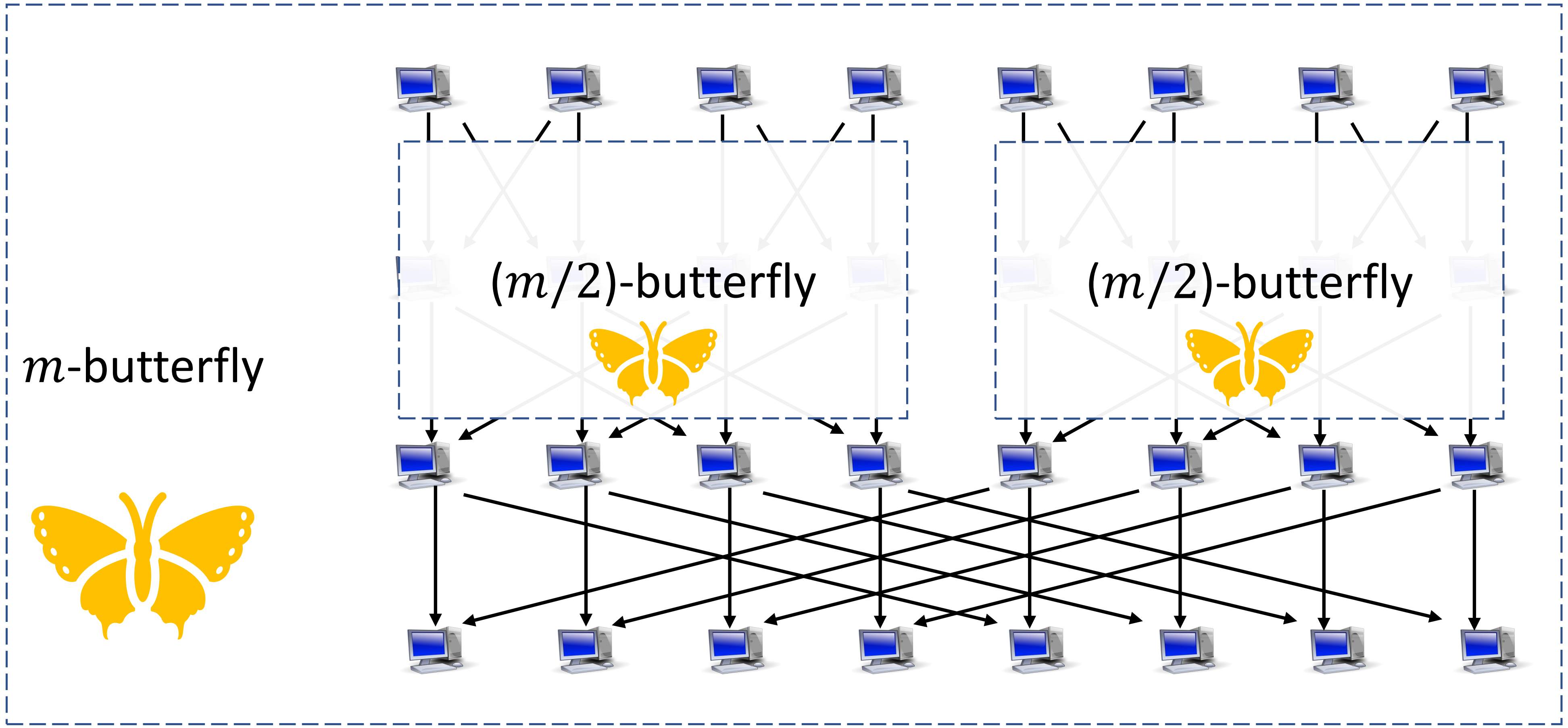
Space $s = N^{\epsilon}$

gets sender-receiver info

Want a **Routing:**

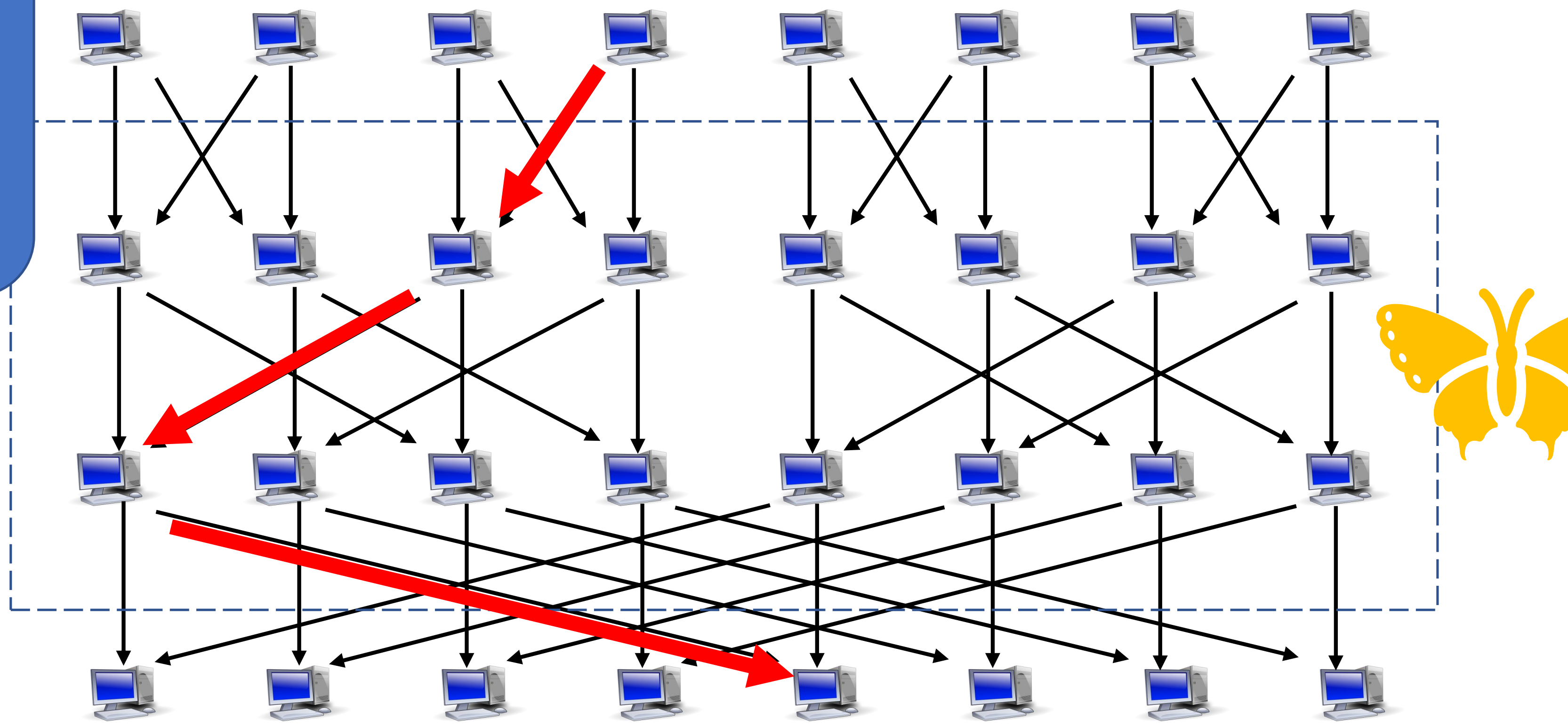


# Butterfly Network (well-known)

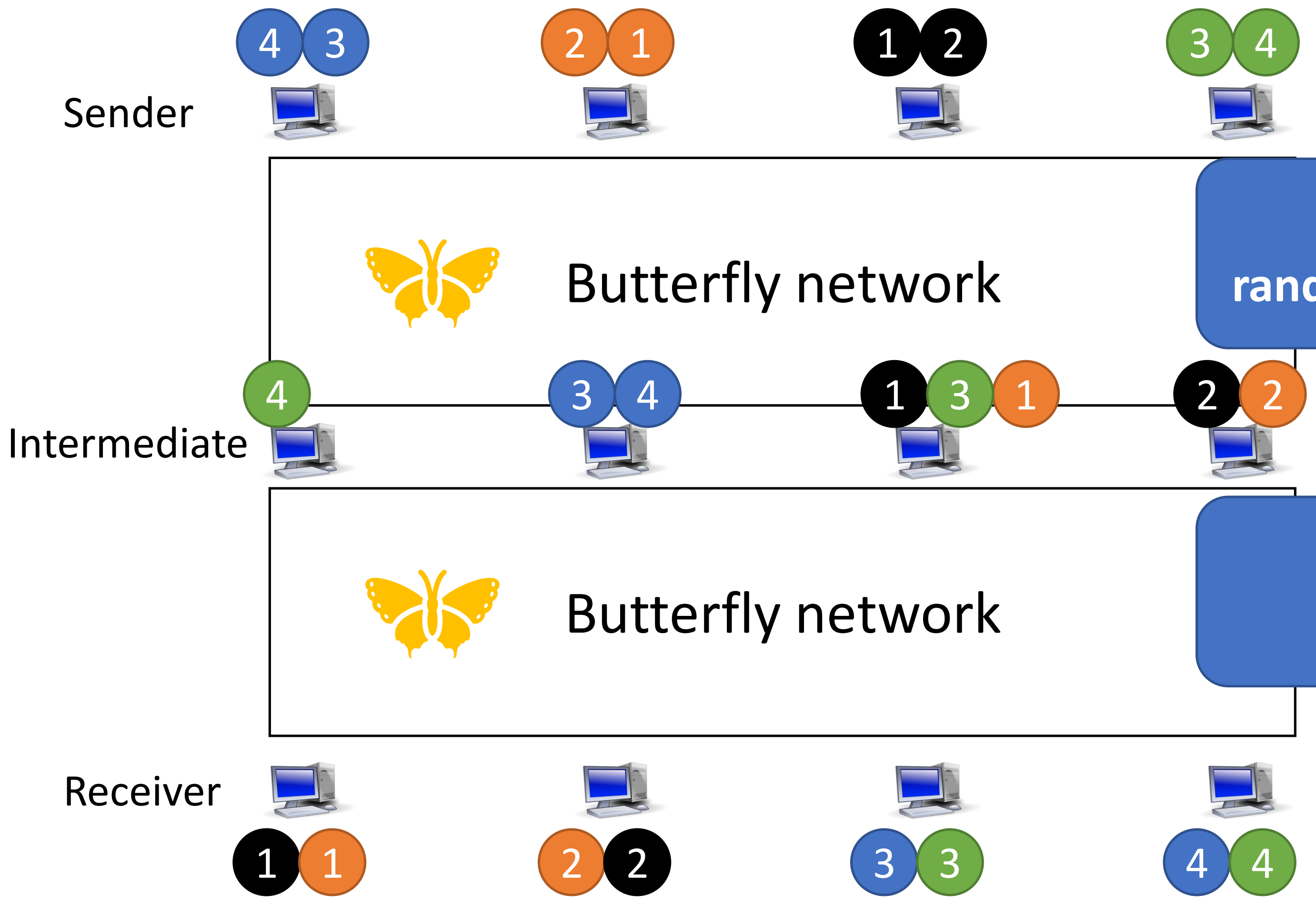


# Butterfly Network (well-known)

Exist a path
for any (sender, receiver),
very easy to find it



# Routing from Butterfly Network

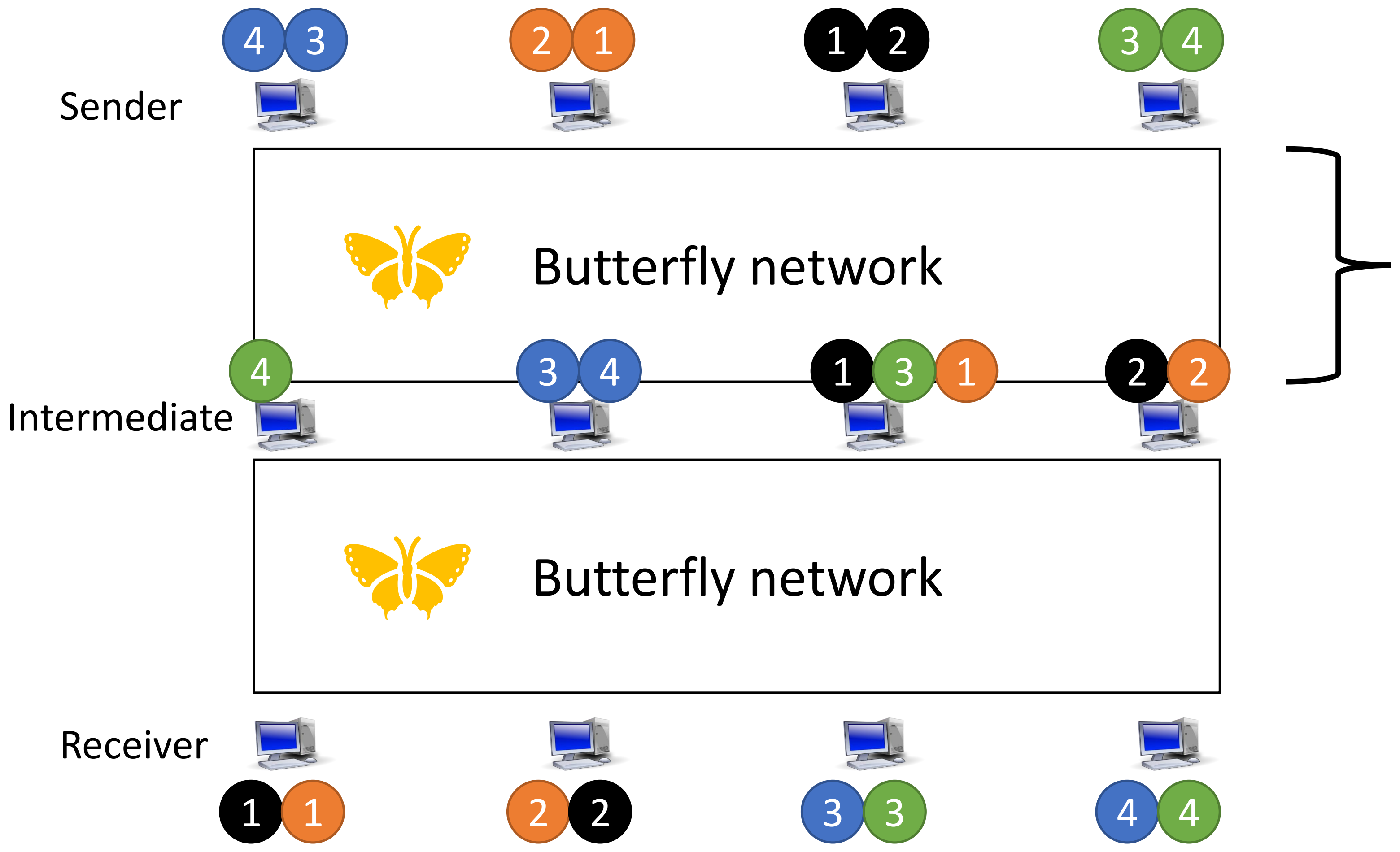


Send to
**random-picked** intermediate

Send to
**prescribed** receiver

Security:
With high probability,
Load on any edge is at most
const-times than expectation

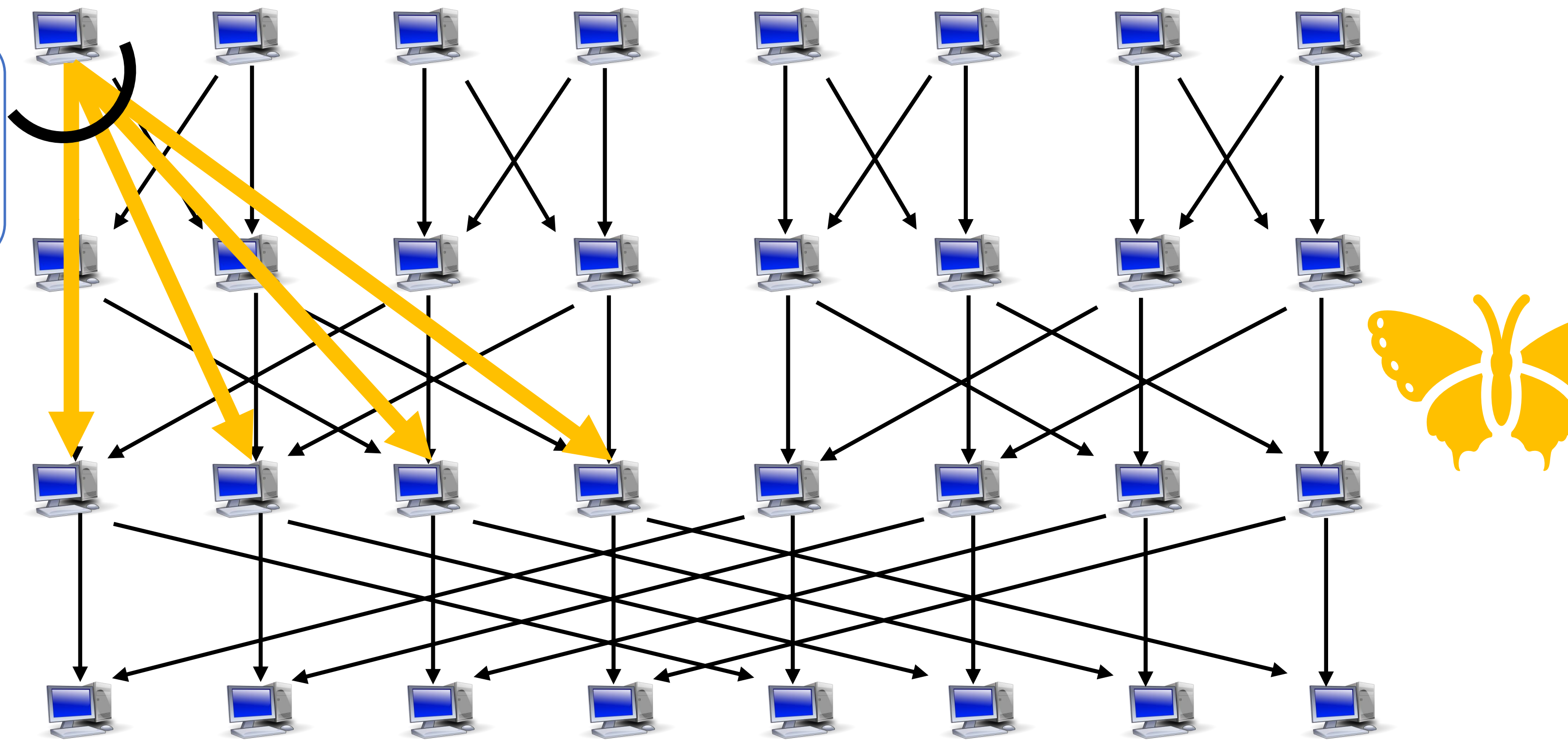
# Routing from Butterfly Network



$\log m$ layers
$\Rightarrow \log m$ rounds ☹

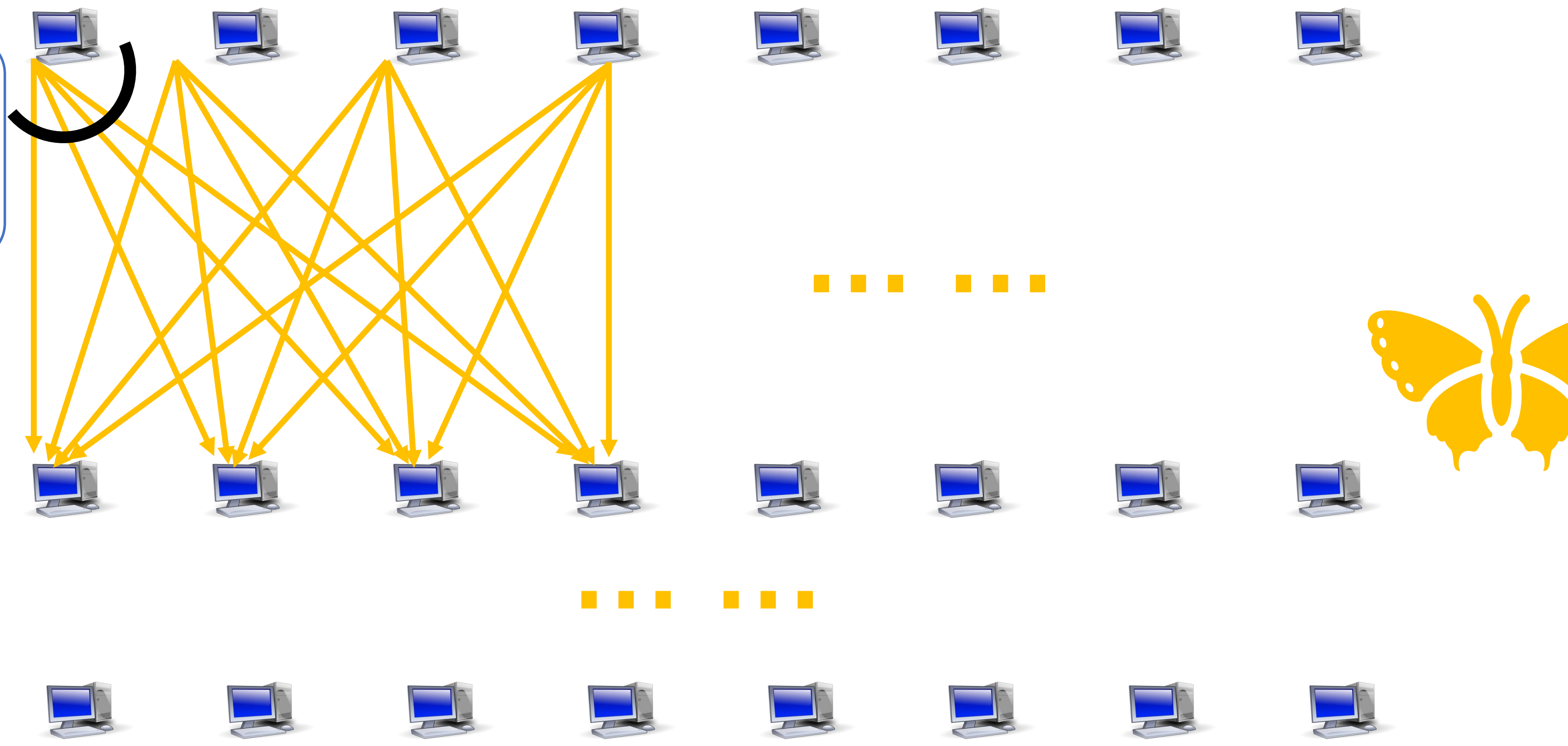
# Idea: Degree $s$ Butterfly Network

Use space $s = N^{\epsilon}$ to merge $\log s$ layers



# Idea: Degree $s$ Butterfly Network

Use space $s = N^{\epsilon}$ to merge $\log s$ layers



$1/\epsilon$ layers

$\Rightarrow$ const rounds ☺

## Summary

Compile insecure Massively Parallel Computation algo into a secure one

Eavesdropping adversary: const overhead in rounds & space

1/3 corrupt machines:
const overhead in rounds, poly(security para) in space

(Need crypto assumptions)

Thank you!

# Previous result and Discuss

Compare to typical secure multiparty computation

- Const rounds, local space ≈ circuit complexity
- Many rounds, smaller local space

Remove crypto assumptions?

- If we can secure any MPC algo using no assumption, then we have a statistical SMPC using small communication (solve open problem)