Asynchronous Secure Multiparty Computation in Constant Time

[PKC'16]

Ran Cohen

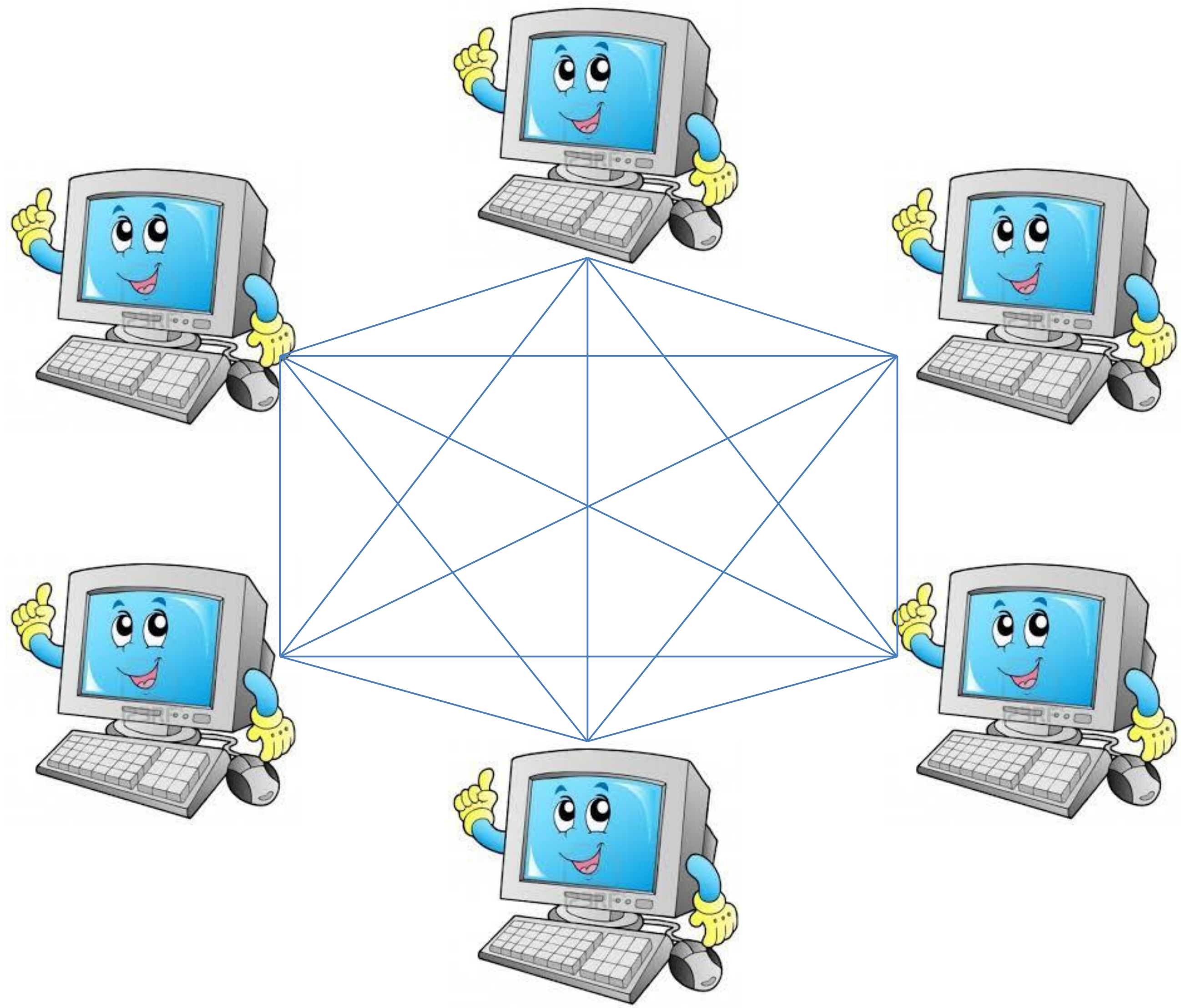
Bar-Ilan University

Information Sharing

- A **terrorist threat** over the world
- Several **intelligence agencies** try to stop it
- Each agency has **secret data** – can't stop attack alone
- If **sufficiently many** agencies join forces – they can stop the attack together
- The terrorists have **double agents** in some agencies
- The terrorists can **delay communication**

Can the world be saved in time?

Secure Multiparty Computation



Security Requirements

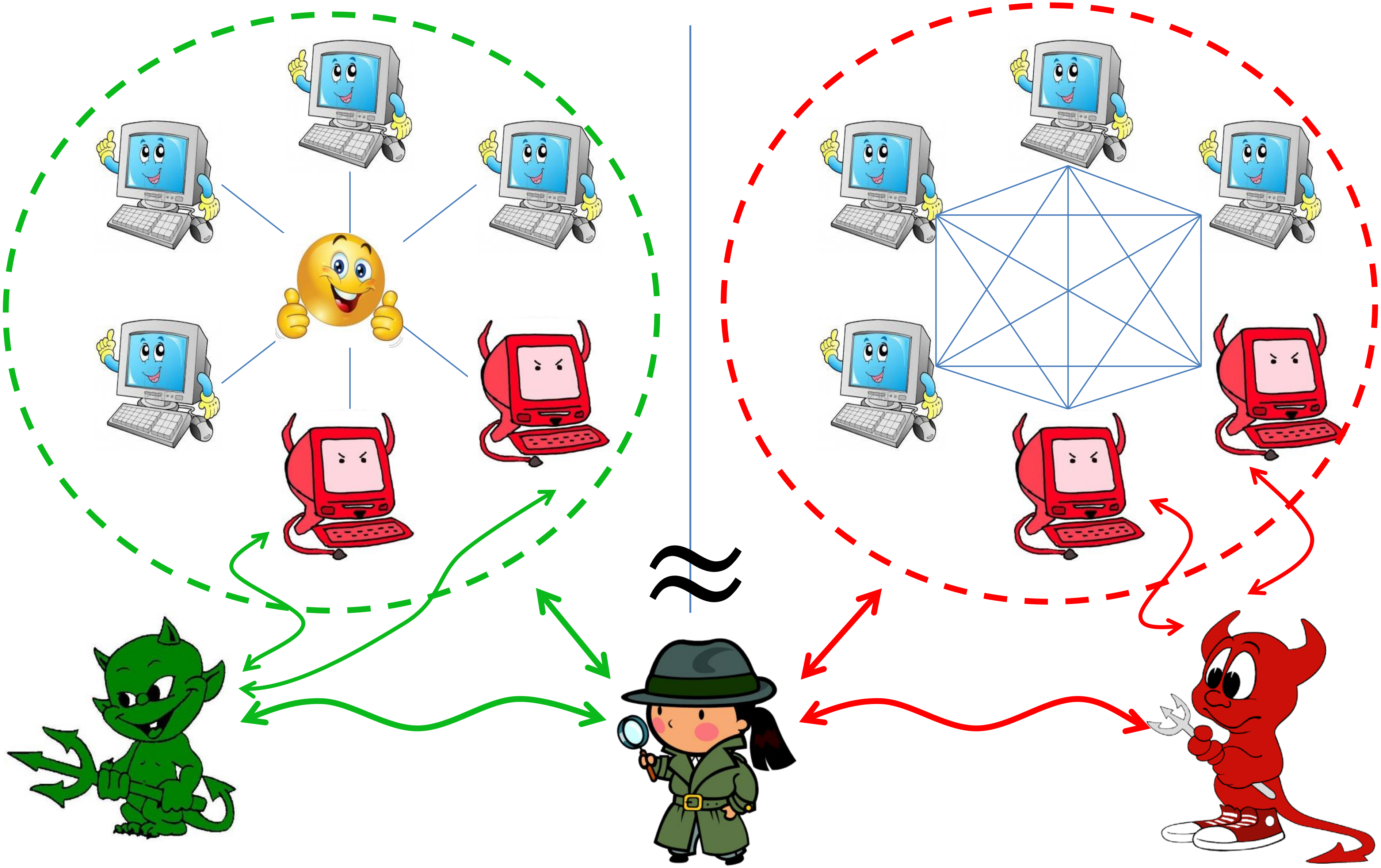
– **Correctness**: parties obtain correct output (even if some parties misbehave)

– **Privacy**: only the output is learned (nothing else)

– **Input completeness**: the inputs of all honest parties are considered in the computation

– **Guaranteed termination**: the computation completes after a finite number of steps

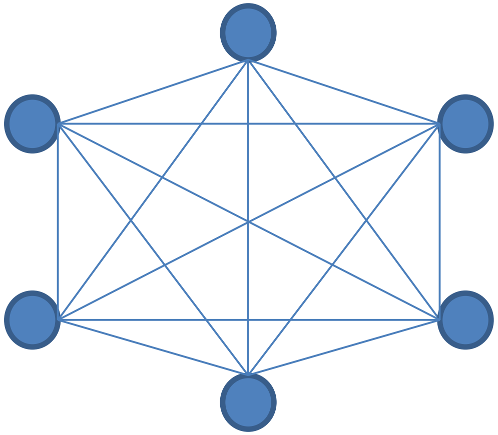
Simulation-Based Security



Communication Model

Point-to-Point (P2P) Model

Authenticated communication lines between every pair of parties



Message delivery:

- Synchronous
- Asynchronous (with eventual delivery)
- Fully-asynchronous (no guaranteed delivery)

Message Delivery

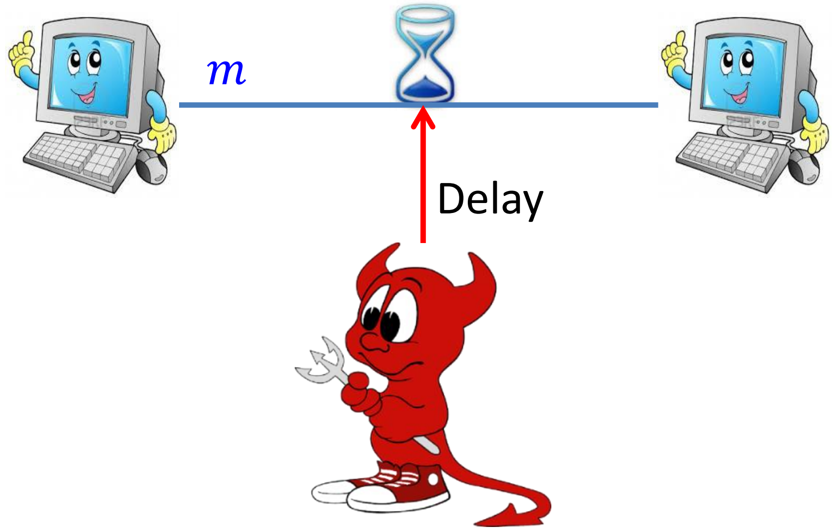
- Synchronous communication
  - Guaranteed delivery (within known time window)
  - Round structure (time-outs)
  - Mainly used in stand-alone setting

- Fully-asynchronous communication
  - $\mathcal{A}$ has **full control** over message delivery
  - Delivery of each message is **not guaranteed**
  - The communication model in UC [Canetti'01]

Asynchronous with Eventual Delivery

- Delivery of each message is **guaranteed**
- $\mathcal{A}$ has control over **timing** of message delivery
- Eventual-delivery channels [KMTZ'13] (arbitrary & finite delay)

Time complexity:
Normalize the maximal delay of a message to 1



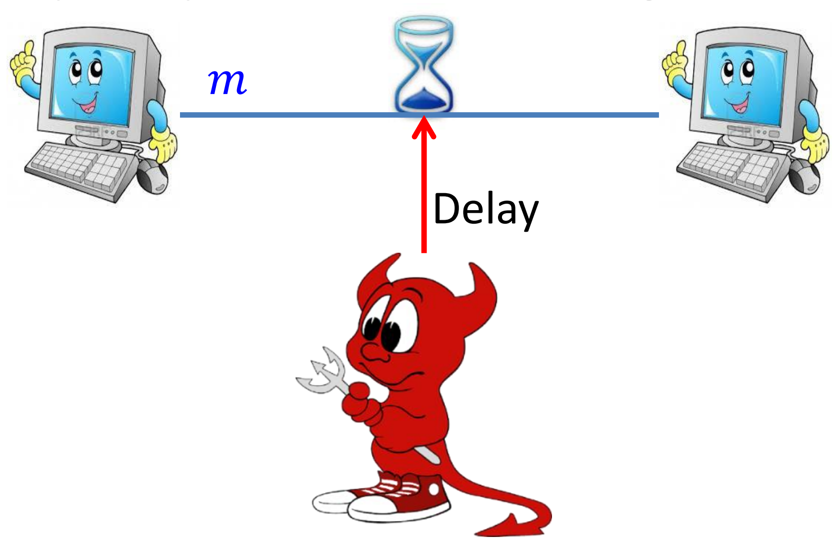
ED-Asynchronous – Main Obstacle

No time-out

Honest parties cannot distinguish between:

1) A corrupted party not sending a message

2) An honest party whose messages are delayed



Asynchronous Byzantine Agreement (ABA)

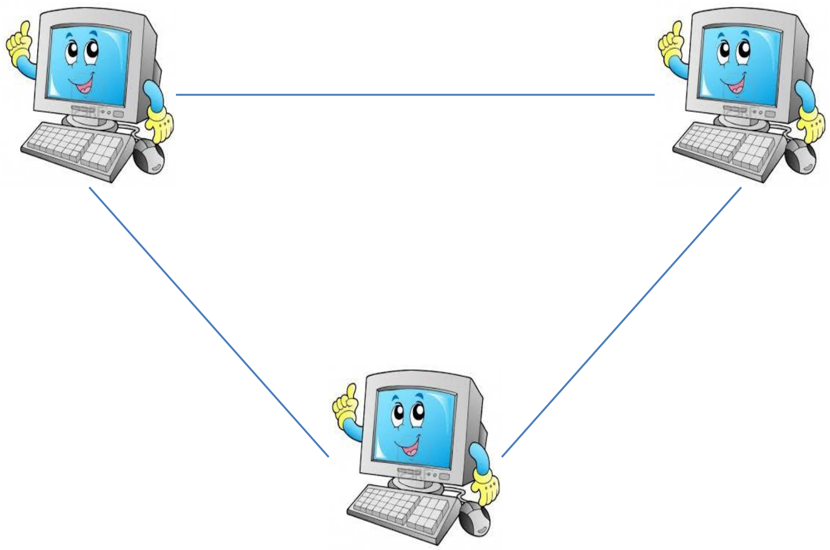
Each party P_i has an input bit $x_i \in \{0,1\}$

- **Agreement**: all honest parties output the same bit
- **Validity**: if all honest parties have the same input, this is the common output

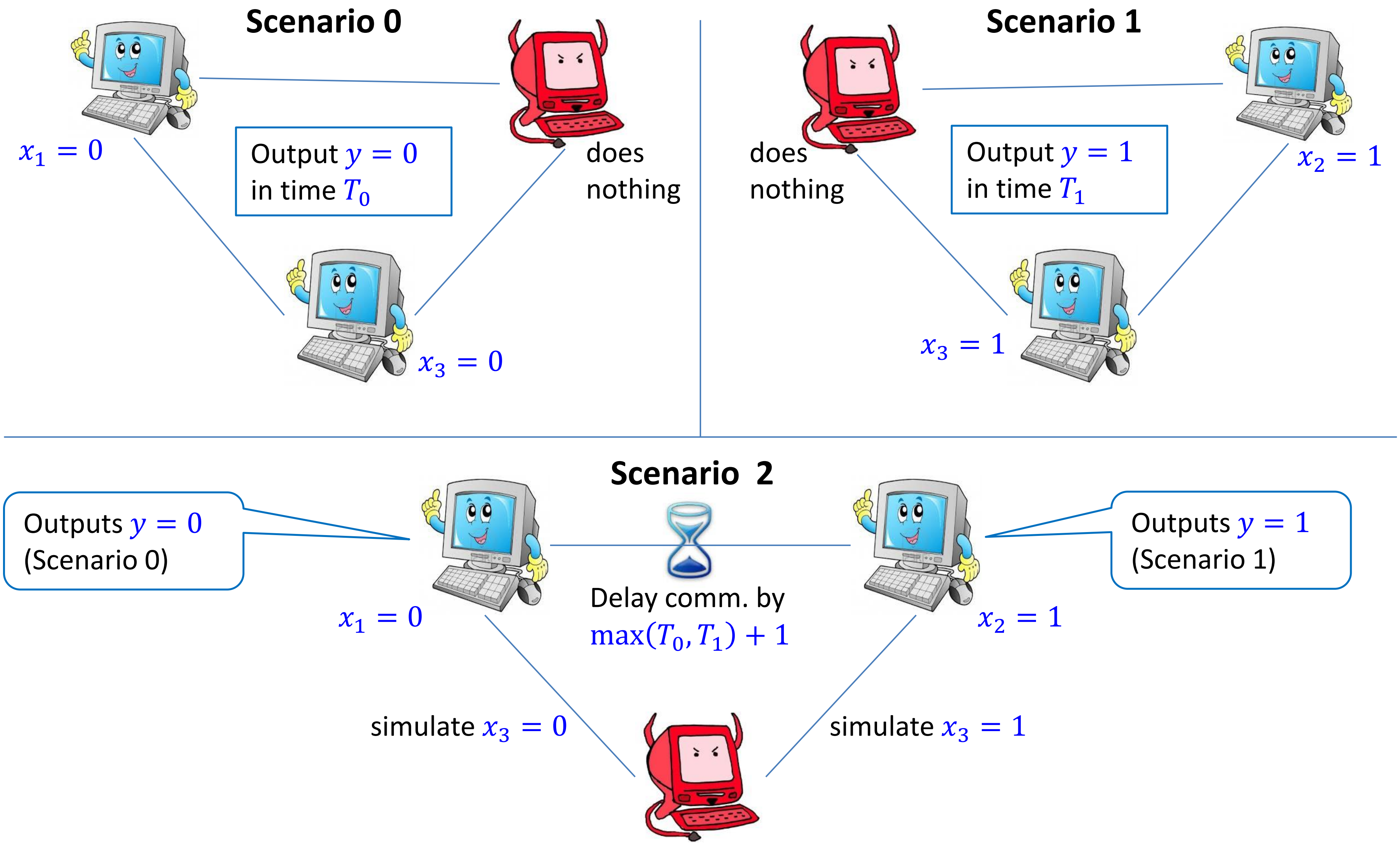
Thm [Toueg'84]: No ABA for $t \geq n/3$ (even with PKI)

<u>Proof</u>

Assume that a 3-party protocol is secure for $t = 1$



Asynchronous Byzantine Agreement (ABA)



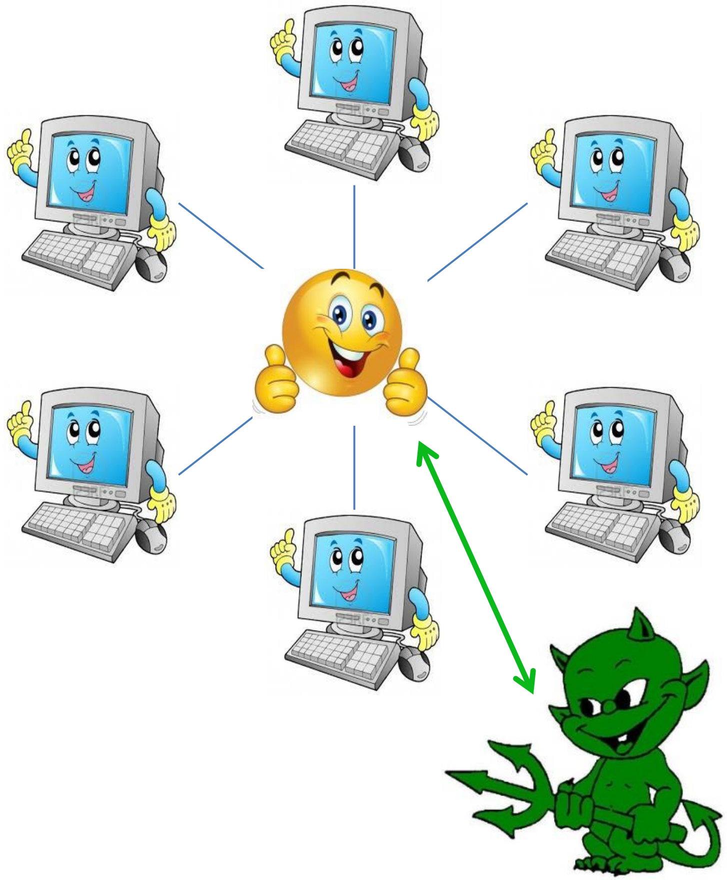
Known Feasibility Results

	Synchronous [GMW'87] [BGW'88]	Fully Asynchronous [CLOS'02]	ED Asynchronous [CLOS'02]	ED Asynchronous [BCG'93] [BKR'94]
Input Completeness	✓	✓	✓	✗
Guaranteed Termination	✓	✗	✗	✓
Constant Time	✓ [BMR'90]	✓ [IPS'08]	✓ [IPS'08]	?
Comm. ind. of f	✓ [AJLTVW'12]	✓ [AJLTVW'12]	✓ [AJLTVW'12]	?

The Ideal Model

No input completeness with **guaranteed termination**:

- $\mathcal{A}$ specifies a core-set C of $n - t$ input providers (t might be corrupted)
- When $\mathcal{T}$ receives inputs for C:
  fix default inputs for $\mathcal{P} \setminus C$
  compute $y = f(x)$
  prepare (y, C) as output
- Each party requests the output from $\mathcal{T}$
- $\mathcal{A}$ can instruct $\mathcal{T}$ to ignore an arbitrary (polynomial) number of requests from P_i



Our Results

Theorem:

Assuming threshold signatures and threshold FHE:

1) There exists a **constant-time** AMPC protocol in the ABA-hybrid model, for $t < n/2$
   Communication complexity independent of the circuit

2) There exists an **expected constant-time** AMPC protocol, for $t < n/3$

No constant-time protocols [FLP'85]

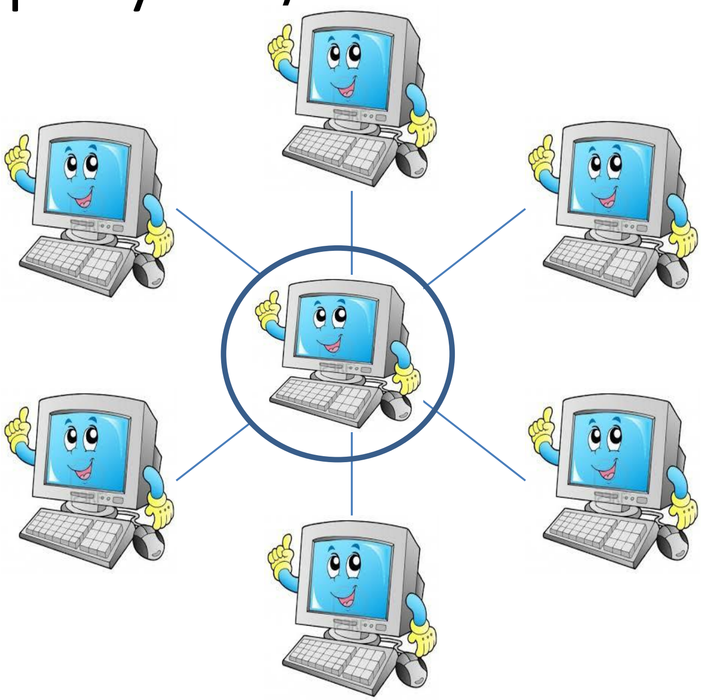
(2) follows from (1) using the concurrent ABA protocol of [BE'03]

Warmup – Multiparty ZKP

A prover P proves a statement x to all other parties

– **Threshold signatures**: sk is $(n-t)$-out-of-n secret shared ($n-t$ signature shares are needed to sign)

1) P proves x to each party V_i (using 2-party ZKP)
2) Once V_i accepts the proof signs a share σ_i for <x is valid>
3) V_i proves to P that σ_i valid (2-ZKP)
4) Upon receiving $n-t$ valid shares P reconstructs signature σ
5) σ is a non-interactive proof for x



The Protocol (Builds on [HNP'08])

Threshold FHE: sk is $(t+1)$-out-of-n secret shared ($t+1$ decryption shares are needed to decrypt)

- Pre-process: key distribution

  Distribute keys for threshold signatures and threshold FHE schemes

1) Input-distribution phase
2) Computation and threshold-decryption phase
3) Termination phase

Input-Distribution Phase

Goal: agree on a core-set of $n - t$ input providers and their encrypted inputs

1) Each P_i computes $c_i \leftarrow Enc_{pk}(x_i)$ and proves to all parties knowledge of the plaintext

2) P_i collects valid proofs from $n - t$ parties $A_i = \{P_{i_1}, \dots, P_{i_{n-t}}\}$, and sends the set A_i to all the parties

3) P_i collects $n - t$ such sets $\{A_{j_1}, \dots, A_{j_{n-t}}\}$, denotes $A = \cup A_j$

4) For every $k \in [n]$ run ABA with input 1 iff $P_k \in A$

5) Let w_k be the kth ABA result. Set $C = \{P_k \mid w_k = 1\}$

Computation and Threshold Decryption

Goal: evaluate the circuit and decrypt result

1) Party P_i sets default inputs for $\mathcal{P} \setminus C$ and evaluates the circuit over $\{c_j\}_{j \in C}$, obtaining $\tilde{c}$
2) P_i decrypts $\tilde{c}$ (obtains share of the output) distributes to all parties proves correctness
3) When P_i collects $\mathrm{t}+1$ valid decryption shares, reconstructs the output y
4) Next, P_i distributes y and proves correctness

Termination Phase

Goal: ensure termination of all honest parties

(After P_i obtains output he must assist proving other parties' statements)

Using **Bracha-style termination**:

- When P_i receives $t+1$ messages for the output y with a valid proof, it accepts y and forwards the proof
- When P_i receives $n-t$ messages for the output y with a valid proof, it terminates

Summary

1) Constant-time AMPC in ABA-hybrid for $t < n/2$

2) Expected constant-time AMPC for $t < n/3$

- Communication complexity independent of f