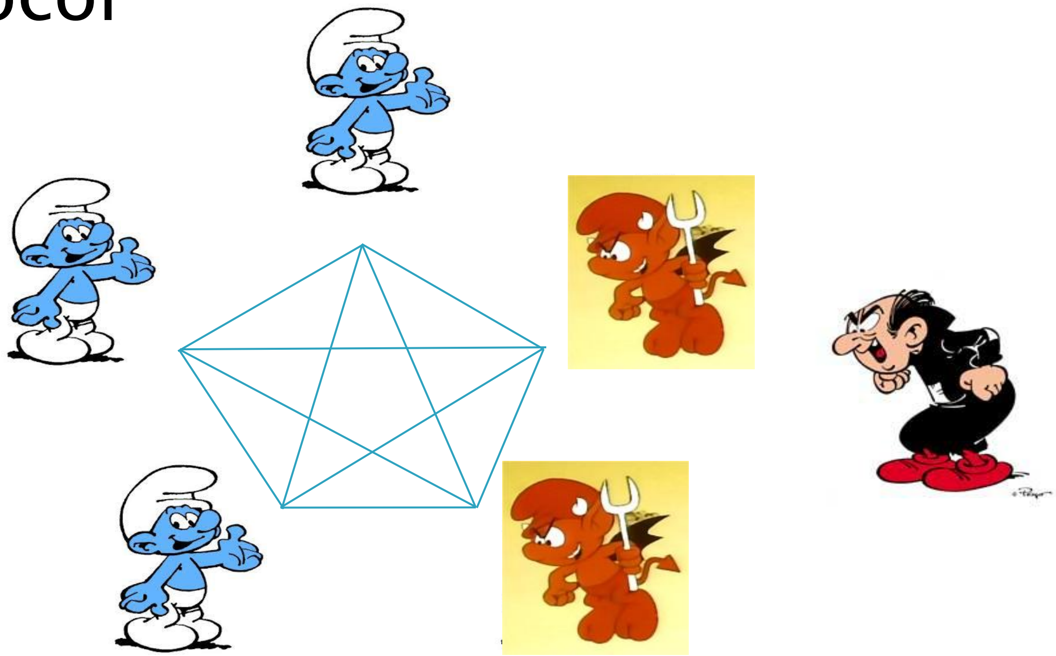
Fairness Versus Guaranteed Output Delivery in Secure Multiparty Computation

Ran Cohen, Yehuda Lindell

Bar Ilan University, Israel

Secure Multiparty Computation

- A set of parties wish to jointly and securely compute a function on their private inputs e.g., voting, auction, etc.
- Security must hold even if some of the parties attack the protocol



Secure Multiparty Computation

- What are the desired security properties?
  - **Correctness:** parties obtain correct output
  - **Privacy:** only the output is learned (nothing else)
  - **Independence of Inputs:** parties cannot choose their inputs based on inputs of other parties
  - **Fairness:** if one party learns the output then all parties learn the output
  - **Guaranteed Output Delivery (G.O.D.):** all parties learn the output

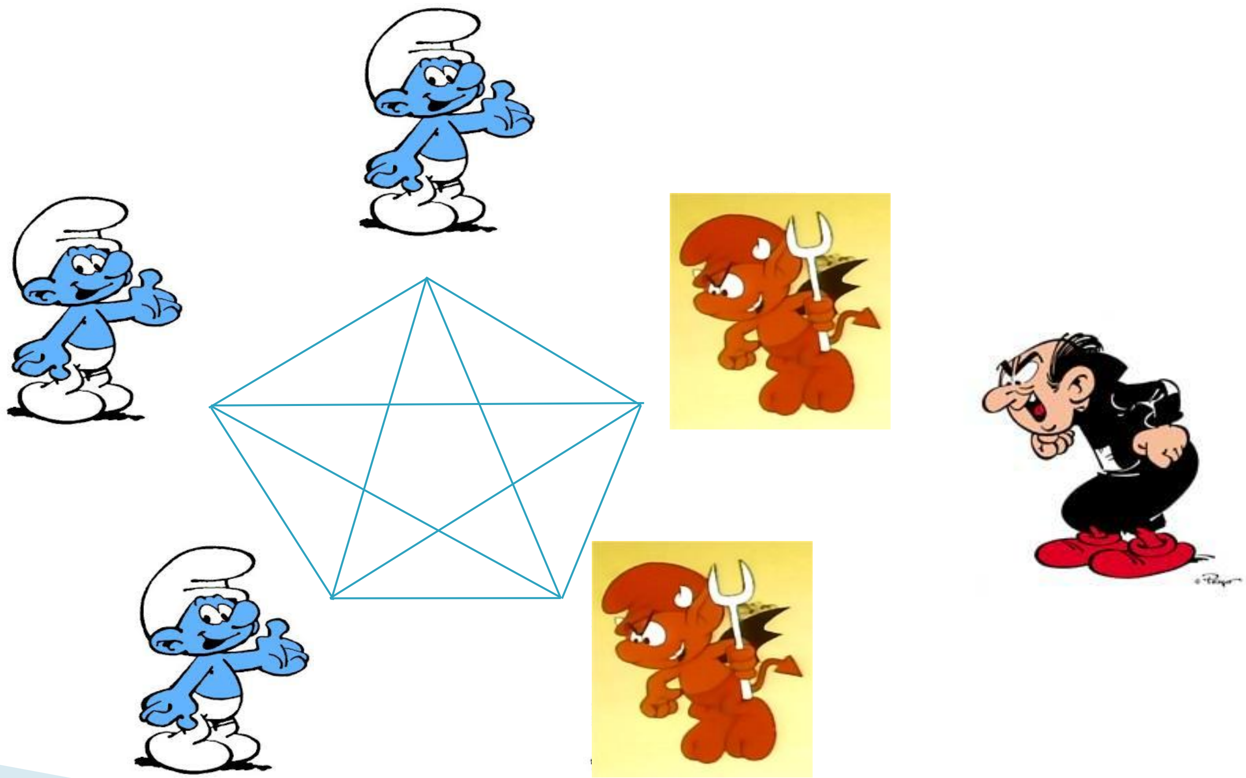
Fairness ⟺ vs. ⟺ G.O.D.

Fairness vs. G.O.D.

Fairness	G.O.D.
If one party obtains output then all parties obtain output	All parties obtain output

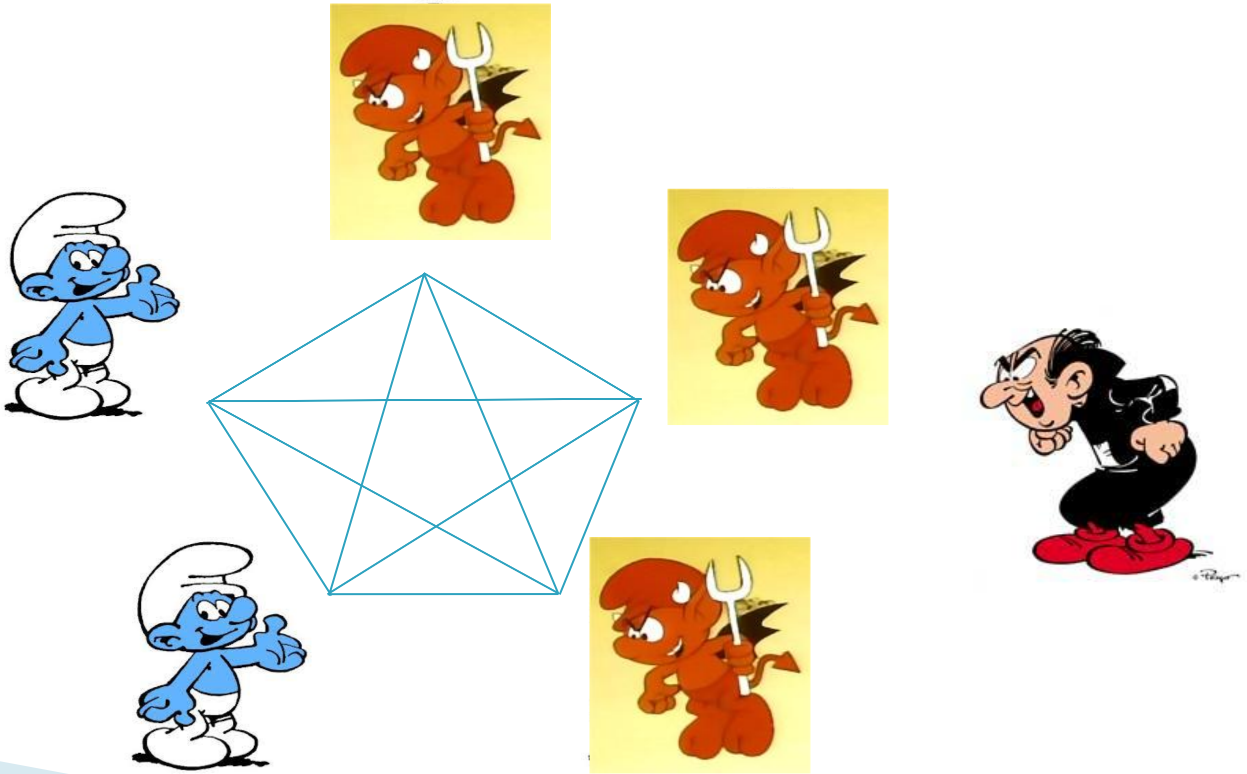
What Do We Know

- Honest majority
  - Every f can be computed with fairness & G.O.D. [GMW87,RB89]



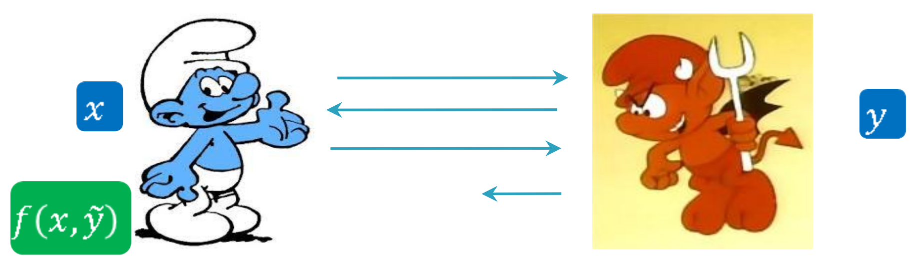
What Do We Know

- No honest majority
  - Fairness & G.O.D. are not always possible [Cleve86]



What Do We Know

- Always
  - G.O.D. ⇒ Fairness
- Two parties
  - Fairness ⇒ G.O.D.
  - In case of (fair) abort, the honest party computes the function locally to obtain output
  - The corrupted party does not learn anything



What Do We Know

Folklore:
Fairness ⇔ G.O.D.

Starting Point

- The broadcast functionality forms a separation between fairness and G.O.D.
- Can be computed with G.O.D. $\Leftrightarrow t < n/3$ [PSL80,LSP82]
- Can be computed with fairness $\forall t < n$ [FGHHS02]
  1) Compute PKI – every party can abort
  2) If abort, fairness is retained – no party learns anything
  3) Else, run authenticated broadcast using the PKI
- However, broadcast is an atypical functionality
  - There is no meaning to privacy
  - Given a secure setup there is no need for cryptography Can be computed $\forall t < n$ information theoretically [PW92]

trivial in the sense of [Kilian91]

Summary of the Results

# Corrupted	Broadcast	P2P
	Fairness & G.O.D. [GMW87,RB89]	
		F; $\exists f$ w Fairness w/o G.O.D.; $\exists f$ w Fairness & G.O.D.
	Fairness ⇔ G.O.D.	in C
	Fairness w Broadcast ⇔ Fairness w/o Broadcast Fail-Stop: Fairness ⇔ G.O.D.	

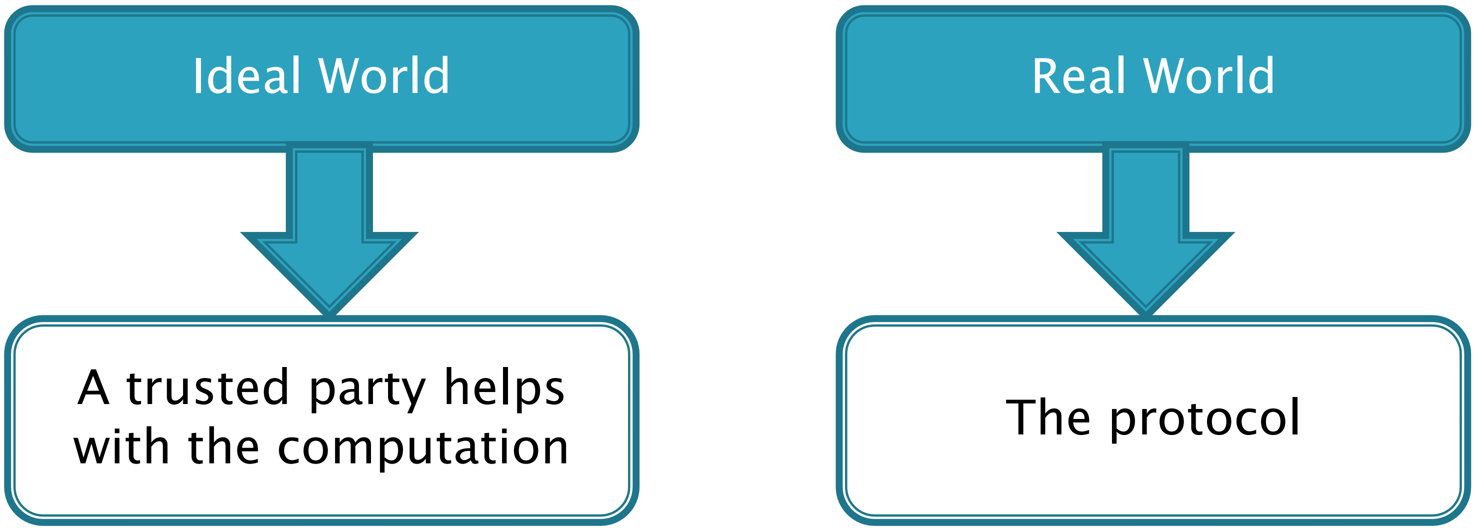
Our Results

Outline

- Some definitions
- Fairness & Broadcast
- Fairness $\not\Rightarrow$ G.O.D.
- G.O.D. & Broadcast
- Conditions for Fairness $\Rightarrow$ G.O.D.
  - Fairness & Broadcast $\Rightarrow$ G.O.D.
  - Fail-Stop: Fairness $\Rightarrow$ G.O.D.

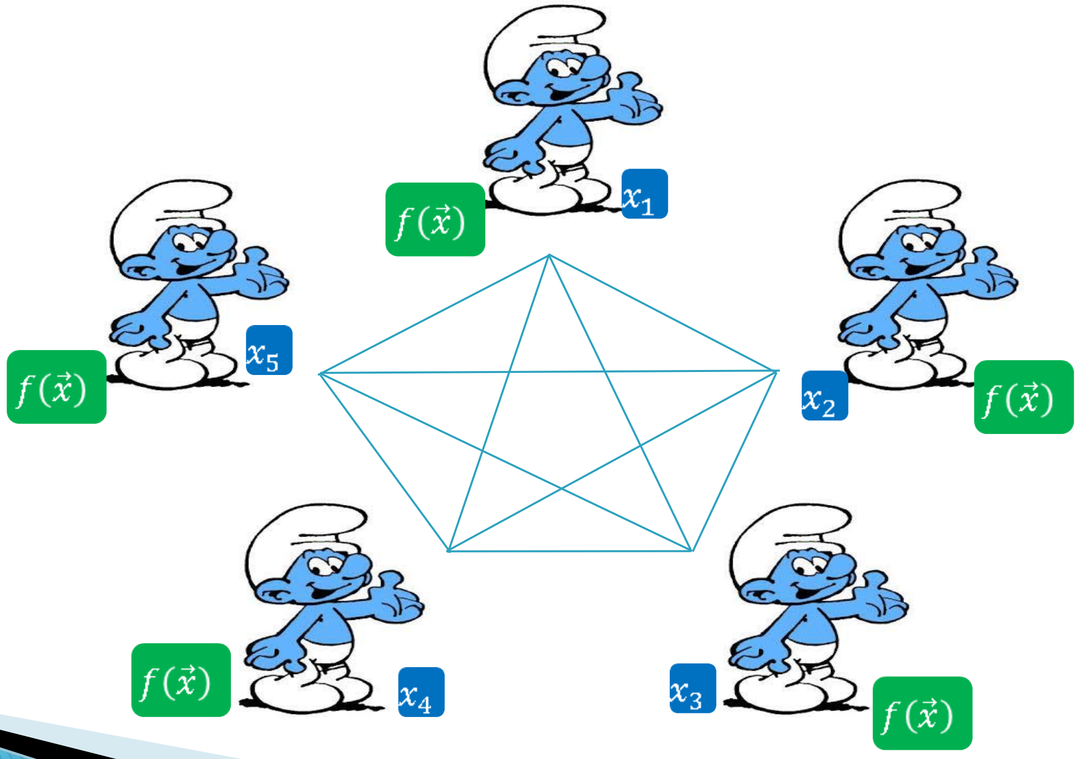
Real/Ideal Paradigm

- The security definition compares two worlds



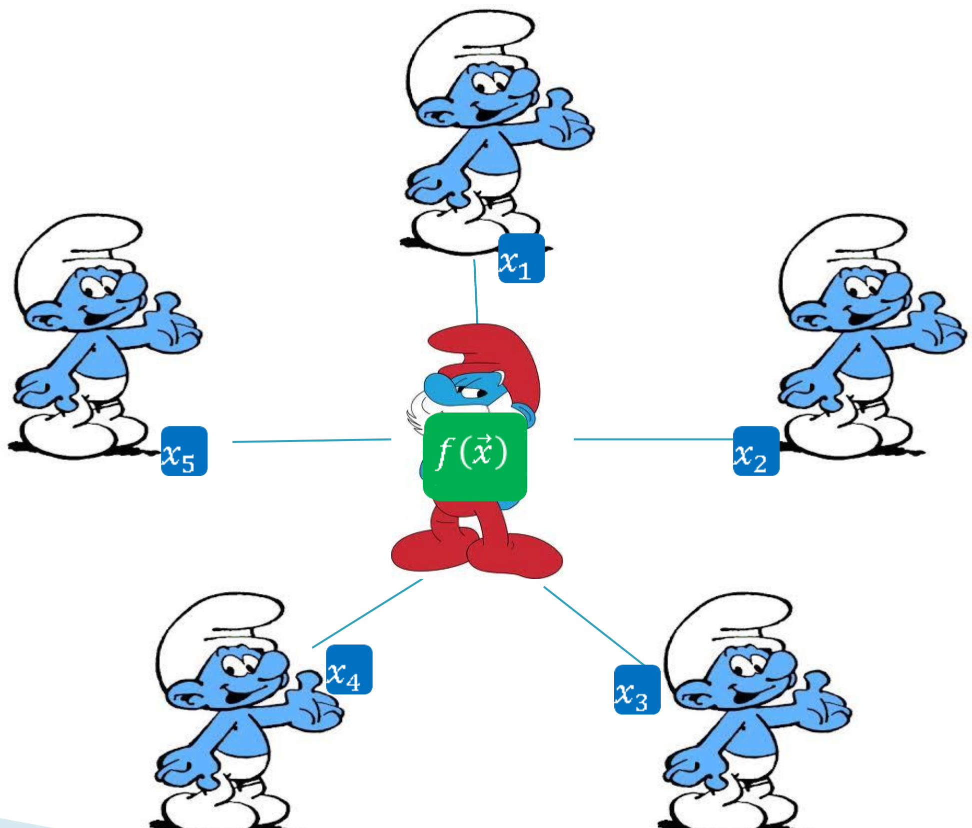
Real World

- Authenticated synchronous network
- Consider either P2P model or broadcast model



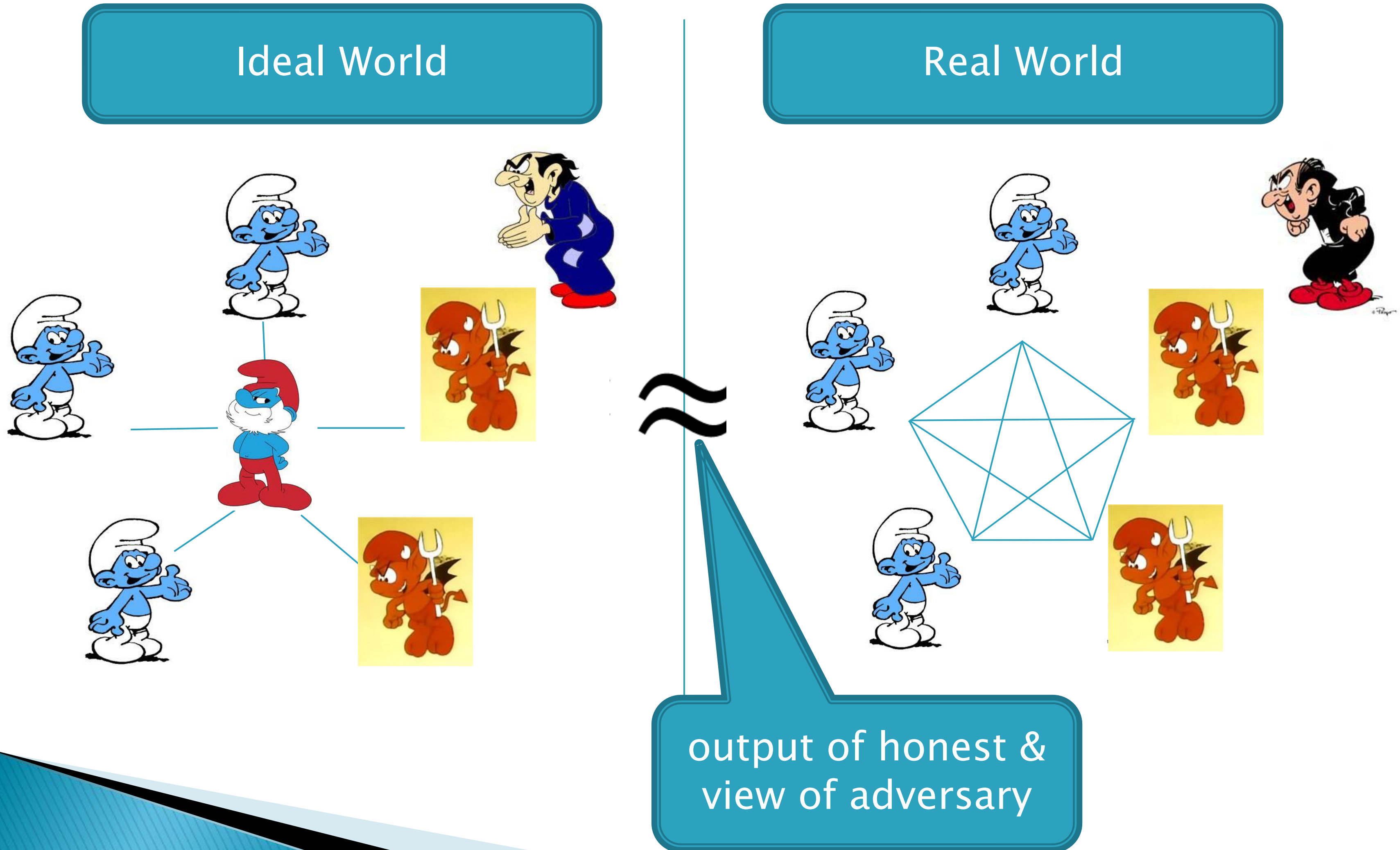
Ideal World

- Trusted party helps computing f



Real/Ideal Paradigm

∀ real $\mathcal{A}$ ∃ ideal $\mathcal{S}$ s.t. the outputs are indistinguishable

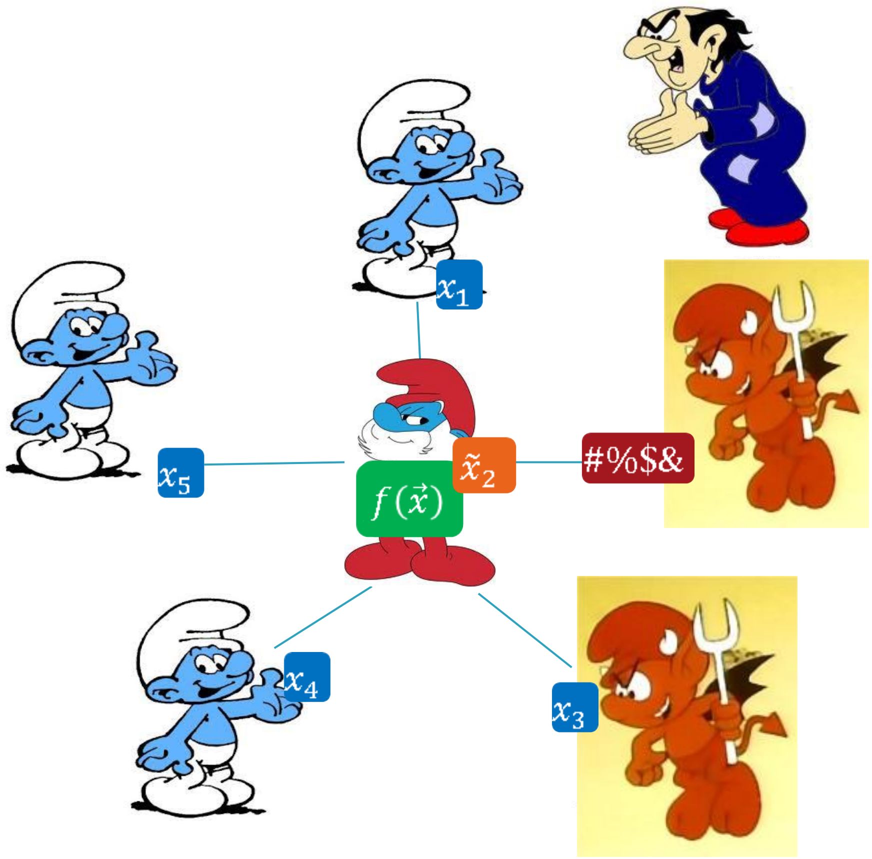


Security of MPC

- Different ideal worlds provide different security:
  - Security with G.O.D.
  - Security with Fairness
  - Security with Fairness and Identified Abort
  - Security with Abort
  - Security with Identified Abort

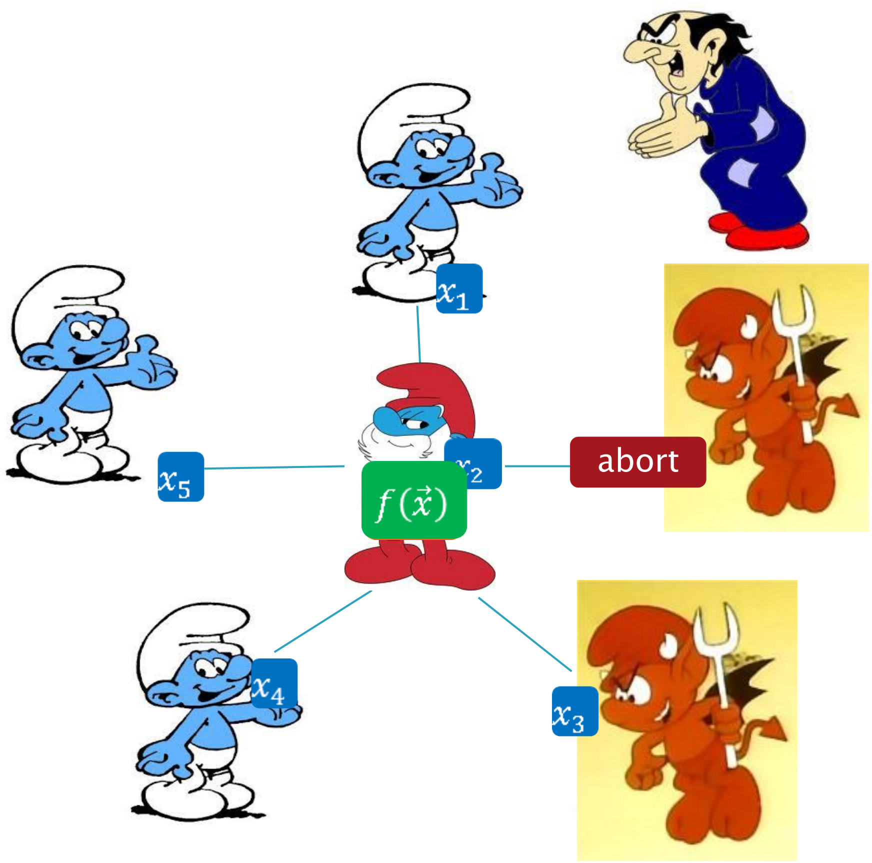
Security with G.O.D.

1. Parties send input to $\mathcal{T}$
2. $\mathcal{T}$ replaces invalid inputs with default
3. $\mathcal{T}$ sends output to parties



Security with Fairness

1. Parties send input to $\mathcal{T}$
2. If $\mathcal{T}$ received abort, send $\bot$ to parties
3. Otherwise, $\mathcal{T}$ sends output to parties
4. Fairness with identified abort: $\mathcal{A}$ can send $(abort, i^*)$ and parties output $(\bot, i^*)$



Outline

- Some definitions
- Fairness & Broadcast
- Fairness $\nRightarrow$ G.O.D.
- G.O.D. & Broadcast
- Conditions for Fairness $\Rightarrow$ G.O.D.
  - Fairness & Broadcast $\Rightarrow$ G.O.D.
  - Fail-Stop: Fairness $\Rightarrow$ G.O.D.

Fairness & Broadcast

Fairness in broadcast model $\Leftrightarrow$ Fairness in P2P model

- Given a fair protocol π for f in broadcast model
- Protocol with fairness for f in P2P model:
  1) Compute PKI with abort as in [FGHHS02]
  2) Run π and replace every broadcast call with authenticated broadcast
- Step (1) is independent of the inputs, so every abort is fair
- Every abort in Step (2) is fair because π is fair

Outline

- Some definitions
- Fairness & Broadcast
- Fairness $\not\Rightarrow$ G.O.D.
- G.O.D. & Broadcast
- Conditions for Fairness $\Rightarrow$ G.O.D.
  - Fairness & Broadcast $\Rightarrow$ G.O.D.
  - Fail-Stop: Fairness $\Rightarrow$ G.O.D.

Separating Fairness & G.O.D.

Goal: $\exists f$ non-trivial with Fairness without G.O.D.

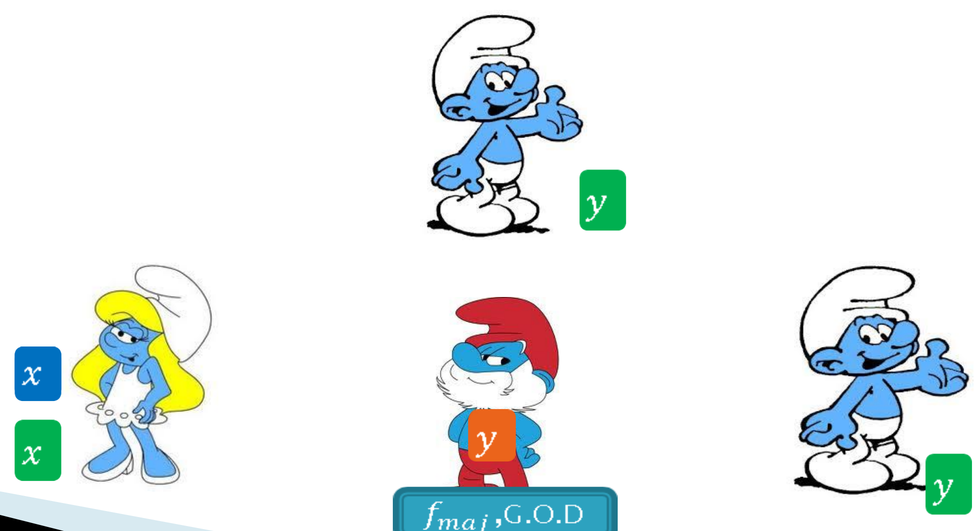
- Recall: Broadcast with G.O.D. in P2P $\Leftrightarrow t < n/3$
- Idea: find non-trivial f that
  - Can be computed with fairness in P2P model
  - If can be computed with G.O.D. then broadcast exists
  - No broadcast $\Rightarrow f$ cannot be computed with G.O.D.
- Three-party majority

$$f_{maj}(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee (x_3 \wedge (x_1 \oplus x_2))$$

- Fair in broadcast model [GK09] $\Rightarrow$ Fair in P2P model
- Non-trivial: 3-party $f_{maj} \Rightarrow$ 2-party OT [Kilian91]

f_{maj} with G.O.D. ⇒ Broadcast

- Consider $\mathcal{T}$ that computes f_{maj} with G.O.D.
- Broadcast protocol in P2P model with $\mathcal{T}$:
  1. Sender sends $x \in \{0,1\}$ to all parties
  2. Each party sends its value to $\mathcal{T}$
  3. Each party gets $y \in \{0,1\}$ from $\mathcal{T}$
  4. Sender outputs x, receivers output y

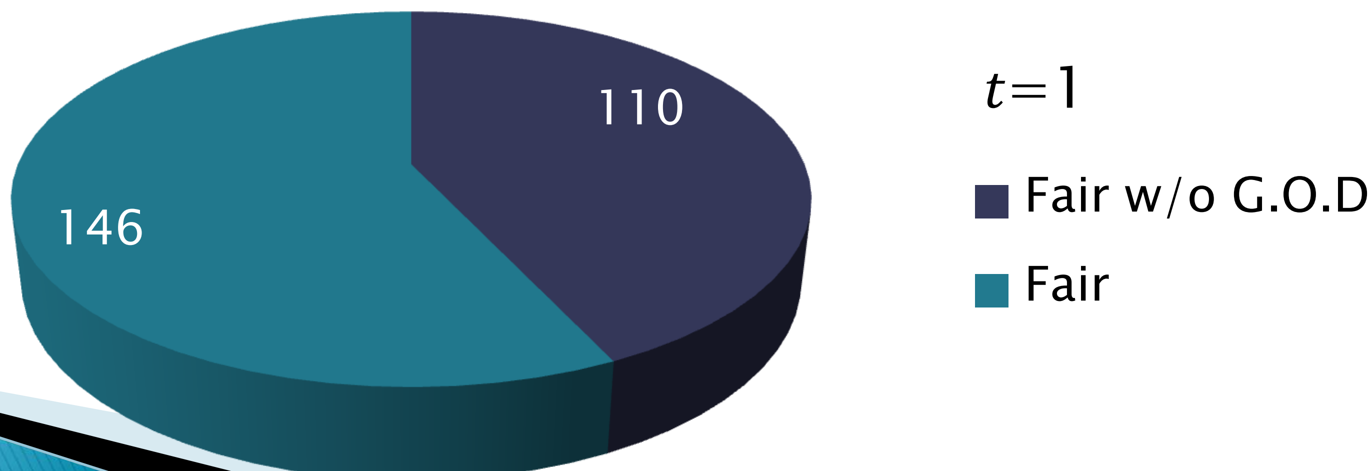


f_{maj} with G.O.D. ⇒ Broadcast

- Intuition for the proof:
  - Corrupted receiver: can send another bit to $\mathcal{T}$
    This doesn't affect the output of f_{maj}
  - Two corrupted receivers: can determine the value y
    This doesn't affect the sender (always outputs x)
  - Corrupted sender: can send different bits
    This doesn't affect consistency of receiver's output
  - Corrupted sender & receiver:
    no affect on honest receiver

Separating Fairness & G.O.D.

- f_{maj} is fair without G.O.D. in P2P model $\forall t < 3$
- We present a sufficient condition for function f to satisfy that f with G.O.D. $\Rightarrow$ broadcast
- 256 functions $f: \{0,1\}^3 \rightarrow \{0,1\}$
  - $t = 1$: 110 imply broadcast $\Rightarrow$ fair without G.O.D.
  - $t = 2$: 8 are fair without G.O.D.



Outline

- Some definitions
- Fairness & Broadcast
- Fairness $\nRightarrow$ G.O.D.
- G.O.D. & Broadcast
- Conditions for Fairness $\Rightarrow$ G.O.D.
  - Fairness & Broadcast $\Rightarrow$ G.O.D.
  - Fail-Stop: Fairness $\Rightarrow$ G.O.D.

G.O.D. & Broadcast

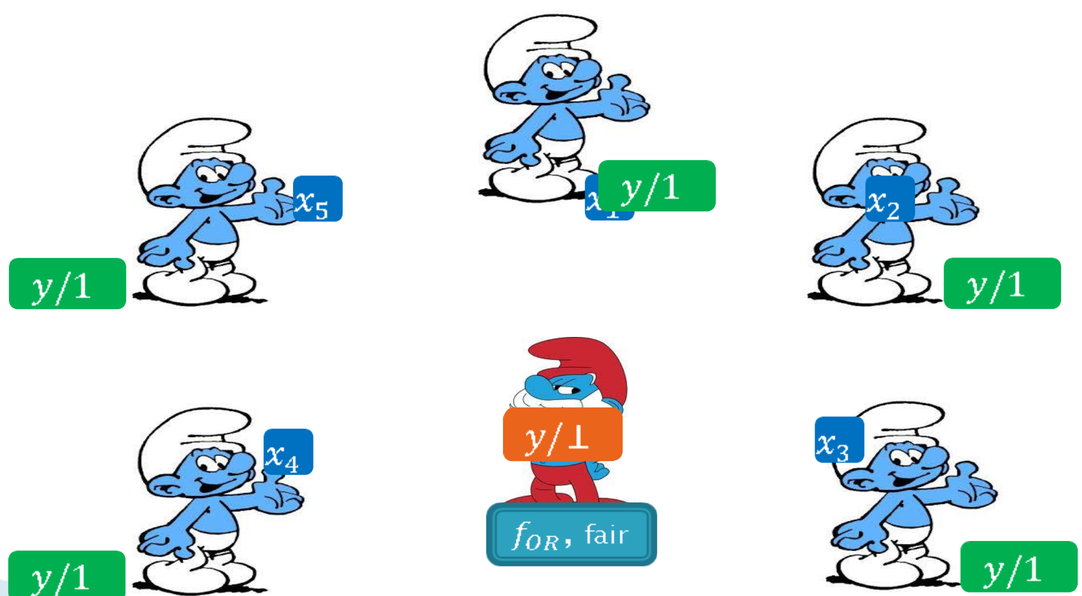
- [GK09] compute f_{maj} & f_{OR} in broadcast model
- f_{maj} cannot be computed with G.O.D. in P2P model
- Is broadcast needed to compute every f with G.O.D?
- No – Multiparty Boolean OR

$$f_{OR}(x_1, \dots, x_n) = x_1 \vee \cdots \vee x_n$$

- Can be computed with G.O.D. in P2P model
- Reason:
  - Fair in P2P model (since fair in broadcast model)
  - Every party can force the output to be 1

G.O.D. Without Broadcast

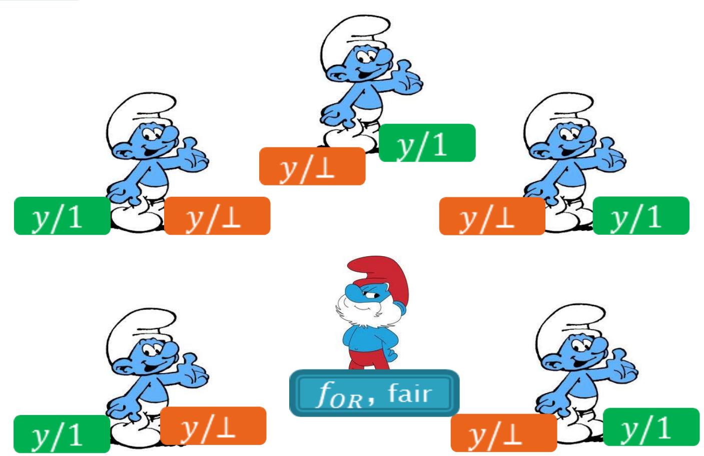
- Consider $\mathcal{T}$ that computes f_{OR} with fairness
- Protocol for f_{OR} with G.O.D. in P2P model & $\mathcal{T}$:
  1. P_i sends x_i to $\mathcal{T}$
  2. P_i receives $y/\perp$ from $\mathcal{T}$
  3. If $y \neq \perp$ P_i outputs y, else P_i outputs 1



G.O.D. Without Broadcast

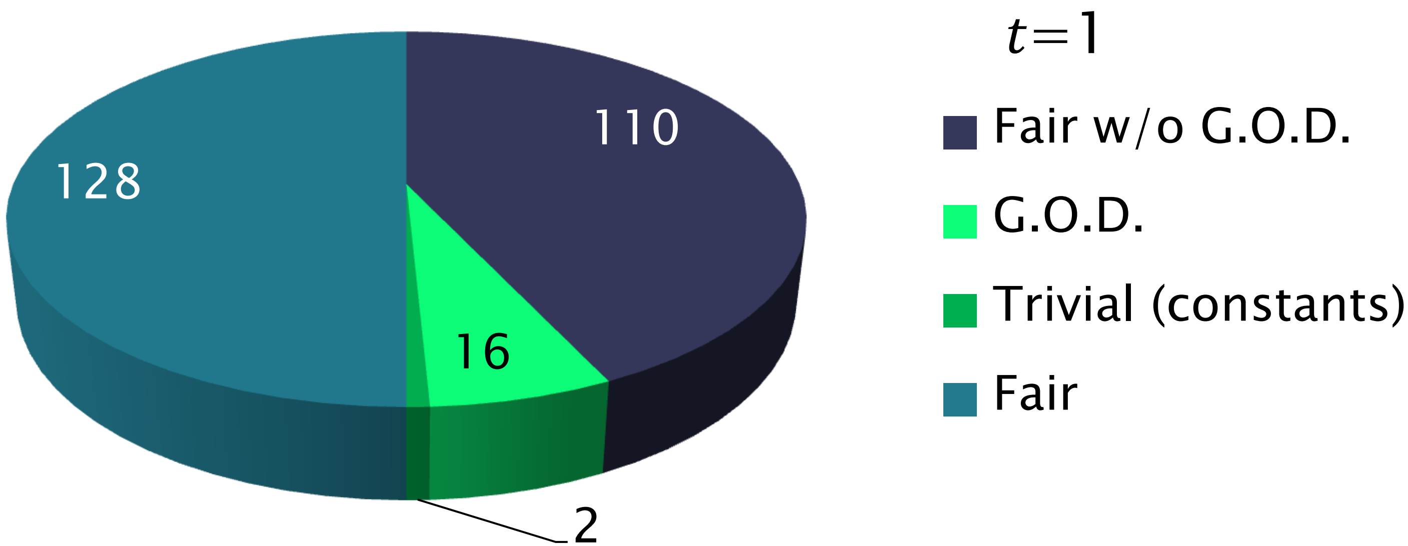
- Intuition for the proof:
  - If $\mathcal{A}$ aborts the protocol, honest parties output 1
  - In this case, $\mathcal{S}$ sends 1 as input in the ideal world
- This idea works for functions where each party can force the output to be some default output

Fairness & Default Output ⇒ G.O.D.



G.O.D. Without Broadcast

- f_{OR} has G.O.D. in P2P model $\forall t < n$
- 256 functions $f:\{0,1\}^3 \to \{0,1\}$
  - 16 are fair with default output $\Rightarrow$ G.O.D. $(t < 3)$



Outline

- Some definitions
- Fairness & Broadcast
- Fairness $\nRightarrow$ G.O.D.
- G.O.D. & Broadcast
- Conditions for Fairness $\Rightarrow$ G.O.D.
  - Fairness & Broadcast $\Rightarrow$ G.O.D.
  - Fail-Stop: Fairness $\Rightarrow$ G.O.D.

When Does Fairness ⇒ G.O.D.

Fairness & Identified Abort ⇒ G.O.D.

- Recall Fairness & Identified Abort:
  If $\mathcal{A}$ aborts:
  - $\mathcal{A}$ does not learn any new information
  - Honest parties learn an identity of a corrupted party
- From fairness & id-abort to G.O.D.
  1) Run the fair protocol
  2) If abort, eliminate a corrupted party and repeat
  3) Else, obtain output and halt
  - Termination after at most $t+1$ iterations

Details in the paper

Fairness & Broadcast ⇒ G.O.D.

- Use GMW compiler with a tweak
- From fairness to fairness & id-abort:
  1) Run π (a fair protocol)
     Every message is proven using ZKP (over broadcast)
  1) If P_i fails to prove a message to P_j – the protocol resumes
  2) When π completes:
     - Either all parties learn the output
     - Or all parties obtain $\perp$ and identify a corrupted party
     - Broadcast : all parties can agree who is cheating

Fail-Stop: Fairness $\Rightarrow$ G.O.D.

- Fail-Stop adversary: can stop sending messages
- From fairness to fairness & id-abort:
  1) Run π (fair against fail-stop)
  2) If P_i didn't send a message to P_j – the protocol resumes
  3) When π completes:
     - Either all parties learn the output
     - Or all parties obtain $\perp$ and P_j identifies P_i as corrupted
  4) Fail-stop: P_j cannot falsely accuse P_i

Summary

- Fairness vs. G.O.D.:
  - Fairness $\not\Leftrightarrow$ G.O.D. in P2P model
  - Fairness $\Leftrightarrow$ G.O.D. in broadcast model
  - Fairness $\Leftrightarrow$ G.O.D. for default output functionalities
  - Fairness $\Leftrightarrow$ G.O.D. for fail-stop adversaries
- Role of Broadcast:
  - Fairness in broadcast model $\Leftrightarrow$ Fairness in P2P model
  - G.O.D. in broadcast model $\not\Leftrightarrow$ G.O.D. in P2P model
- Open questions
  - When Fairness $\Rightarrow$ G.O.D.
  - Old: characterize Fairness New: characterize G.O.D.