Fairness Versus Guaranteed Output Delivery in Secure Multiparty Computation

Ran Cohen (BIU)

Yehuda Lindell (BIU)

Online Poker

Online Poker

Secure multiparty computation

A set of distrusting parties wish to securely compute a joint function of their inputs

- Elections
- Auctions
- Private database search
- Coin flipping
- ...

Security should hold facing an external adversary that controls a subset of the parties

Secure multiparty computation

Security requirements typically include:

- Privacy: only the function output is learned
- Correctness: parties obtain correct output
- And more …

Captured by Real/Ideal paradigm

Hierarchy of security definitions:

- Security with abort: abort **after** obtaining output
- Security with fairness: abort **before** obtaining output
- Security with guaranteed output delivery: **no abort**

What is the difference?

What is the difference?

No Fairness
Adversary may obtain output **BEFORE** the honest parties
In case it is losing – the adversary can abort
Adversary can decide to prematurely abort **BASED ON ITS OUTPUT**

What is the difference?

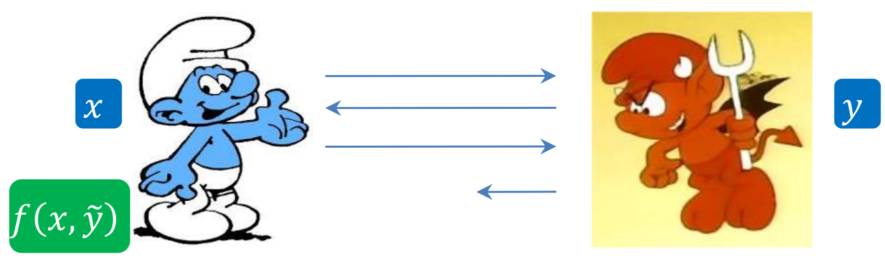
No Fairness	Fairness
Adversary may obtain output **BEFORE** the honest parties	One party obtains output $\Rightarrow$ all parties obtain output
In case it is losing – the adversary can abort	In case it has a bad hand – the adversary can abort
Adversary can decide to prematurely abort **BASED ON ITS OUTPUT**	Adversary can decide to prematurely abort **BASED ON ITS INPUT ALONE**

What is the difference?

No Fairness	Fairness	G.O.D.
Adversary may obtain output **BEFORE** the honest parties	One party obtains output $\Rightarrow$ all parties obtain output	All parties obtain output
In case it is losing – the adversary can abort	In case it has a bad hand – the adversary can abort	Adversary cannot abort under any circumstances
Adversary can decide to prematurely abort **BASED ON ITS OUTPUT**	Adversary can decide to prematurely abort **BASED ON ITS INPUT ALONE**	Denial of Service attacks are **NOT POSSIBLE**

Fairness vs. G.O.D.

- Protocols normally achieve both fairness & G.O.D. or do not achieve neither fairness nor G.O.D.
- G.O.D. $\Rightarrow$ fairness
- Two parties: fairness $\Rightarrow$ G.O.D.
  - In case of (fair) abort, the honest party can locally compute the function using a default input value
  - The corrupted party does not learn anything



Main Question

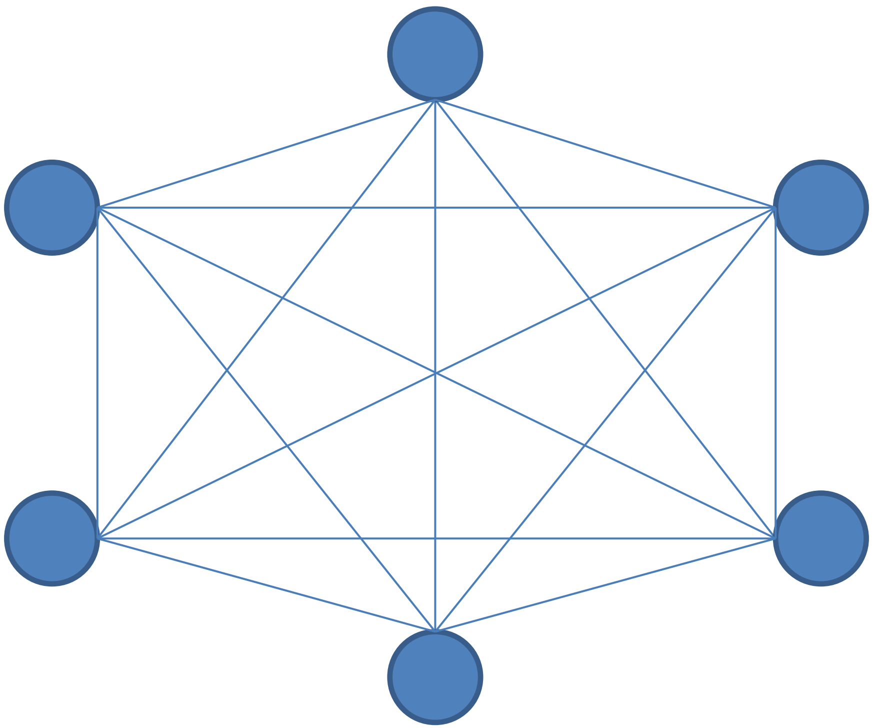
Does fairness imply G.O.D.?

- Are there functions that can be computed with fairness but not with G.O.D.?
- Under which conditions on the network/function does fairness $\Rightarrow$ G.O.D.?

Communication Models

Point-to-Point (P2P)

Authenticated communication lines between every pair of parties

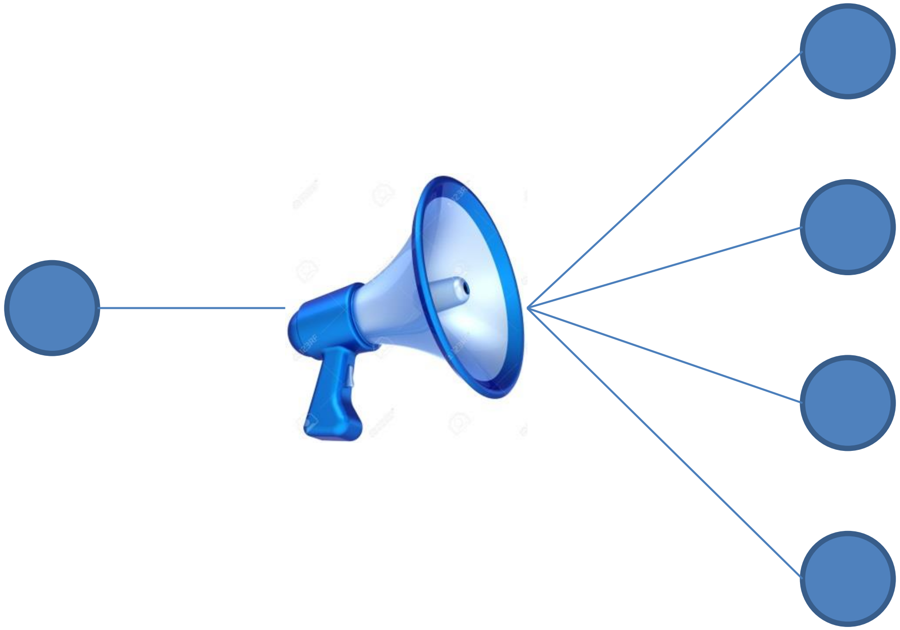


Broadcast channel

When a party sends a message m:

– All honest parties receive the same message m'

– If the sender is honest, then $m = m'$



Feasibility of MPC

Broadcast

- $t < n/2$
  - $\forall f$ G.O.D. (IT) [RB'89]
- $t \geq n/2$
  - $\exists f$ no fairness [Cleve'86]
    - Coin flipping
- $t < n$
  - $\forall f$ security with abort [GMW'87]
  - $\exists f$ G.O.D. [GK'09]
    - Boolean OR
    - Three-party majority

Point-to-Point

- $t < n/3$
  - $\forall f$ G.O.D. (IT) [BGW'88,CCD'88]
- $t \geq n/3$
  - $\exists f$ no G.O.D. [PSL'80]
    - Byzantine agreement
- $t < n/2$
  - $\forall f$ fairness [FGMR'02]
- $t < n$
  - $\forall f$ security with abort [FGHHS'02]
  - $\exists f$ G.O.D. [FGHHS'02]
    - Weak Byzantine agreement

Starting Point

The broadcast functionality separates fairness and G.O.D.

- Can be computed with G.O.D. $\Leftrightarrow t < n/3$ [PSL'80]
- Can be computed with fairness $\forall t < n$ [FGHHS'02]
  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 [PW'92]

trivial in the sense of [Kilian'91]

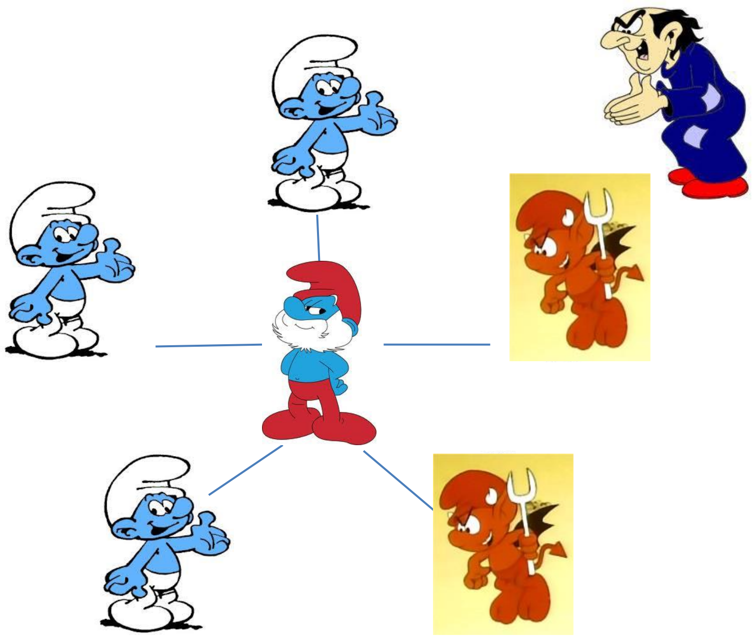
Our Results

- Fairness $\not\Leftrightarrow$ G.O.D. in the P2P model (non-trivially)
  - $\exists f$ can be computed with fairness but not with G.O.D.
- Fairness $\Leftrightarrow$ G.O.D. in the broadcast model
- Broadcast is not necessary for G.O.D.
  - $\exists f$ can be computed with G.O.D. in P2P model
- 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

Real/Ideal Paradigm

Ideal World

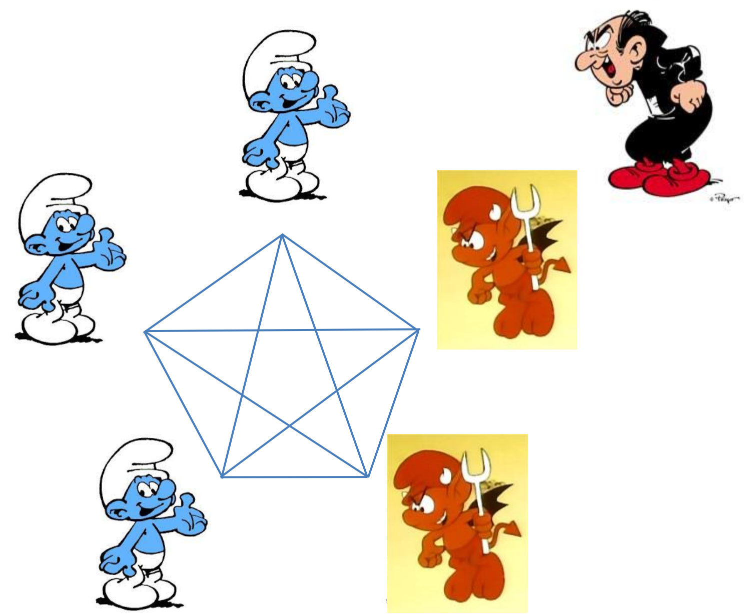
Computing f using a trusted party $\mathcal{T}$



$\approx$

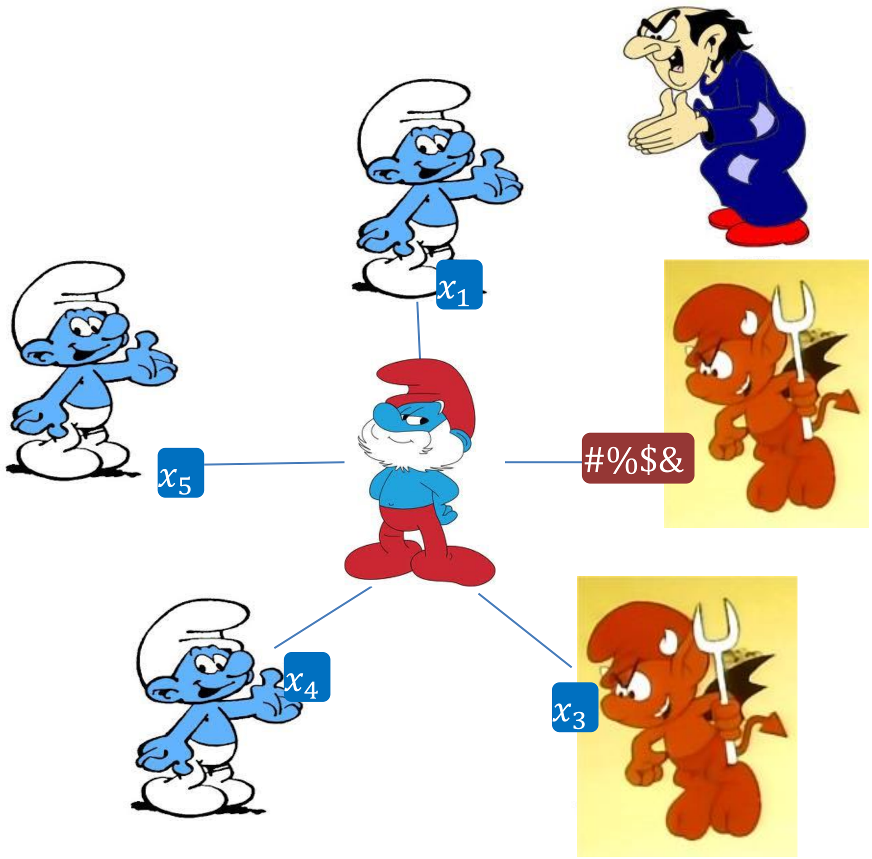
Real World

Computing f with a protocol π



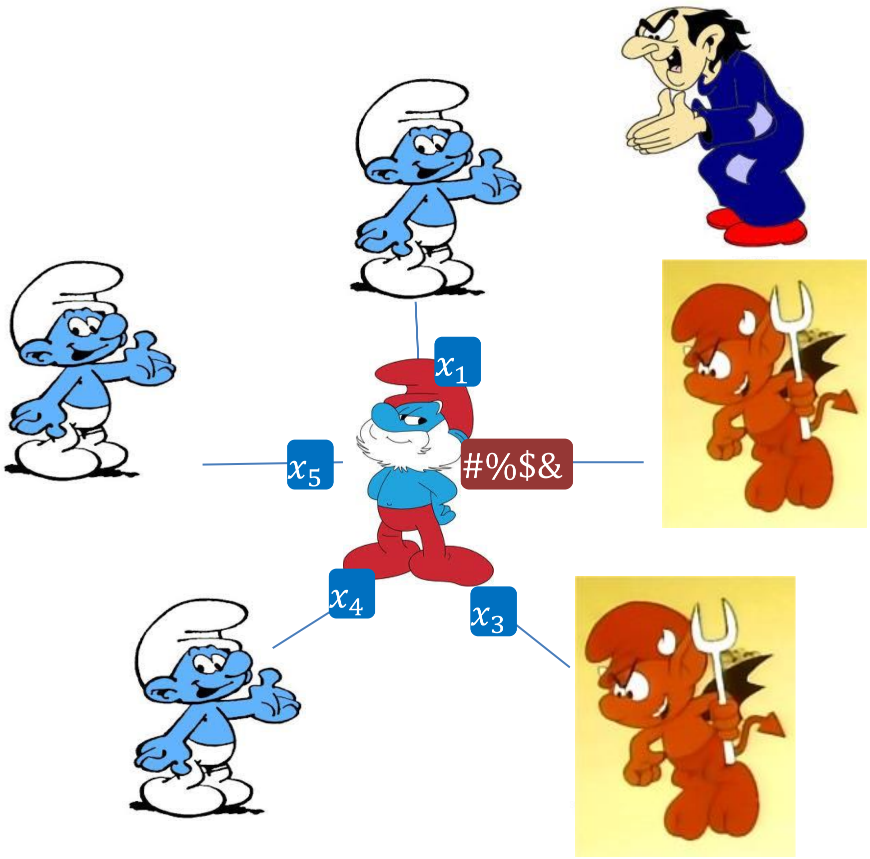
Security with G.O.D.

1. Parties send input to $\mathcal{T}$



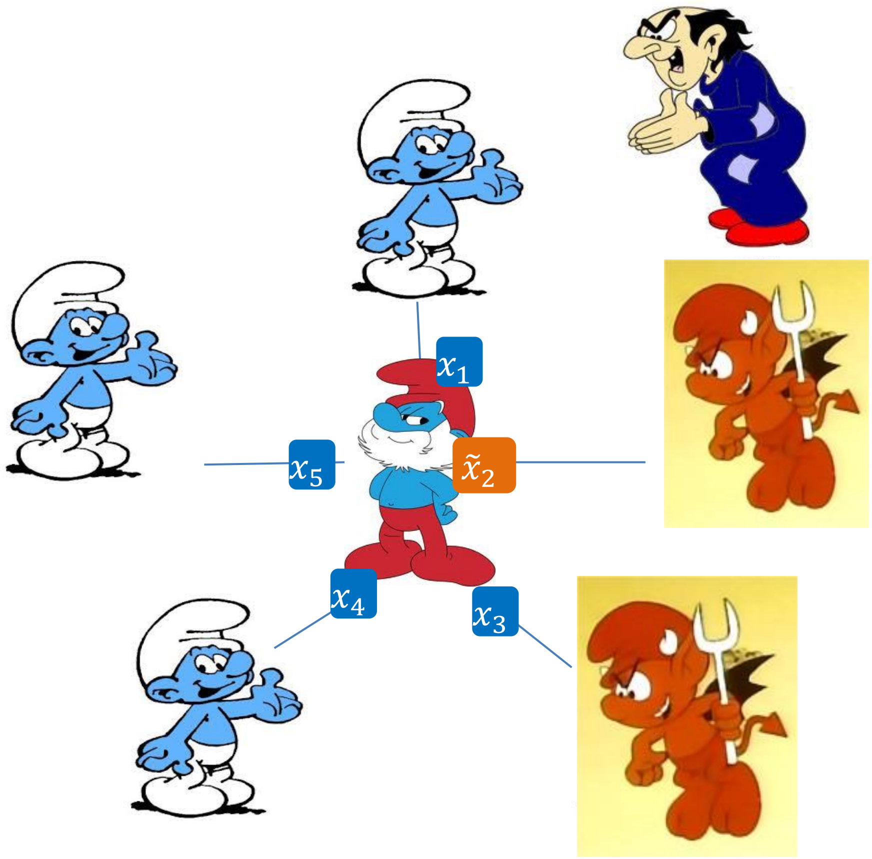
Security with G.O.D.

1. Parties send input to $\mathcal{T}$



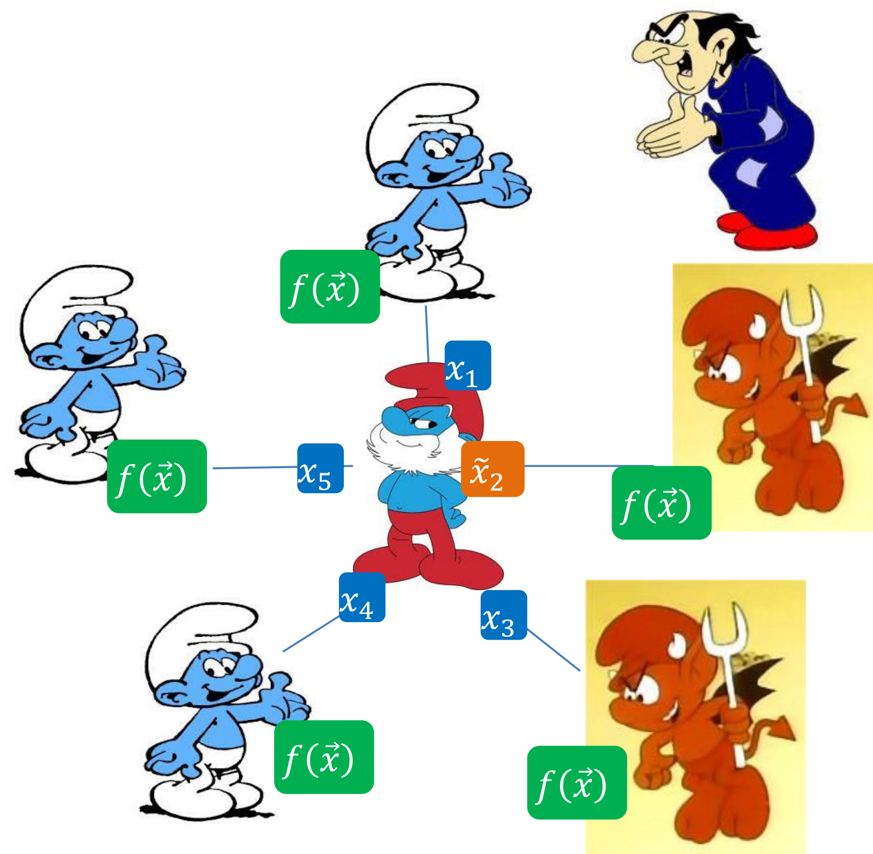
Security with G.O.D.

1. Parties send input to $\mathcal{T}$
2. $\mathcal{T}$ replaces invalid inputs with default input values



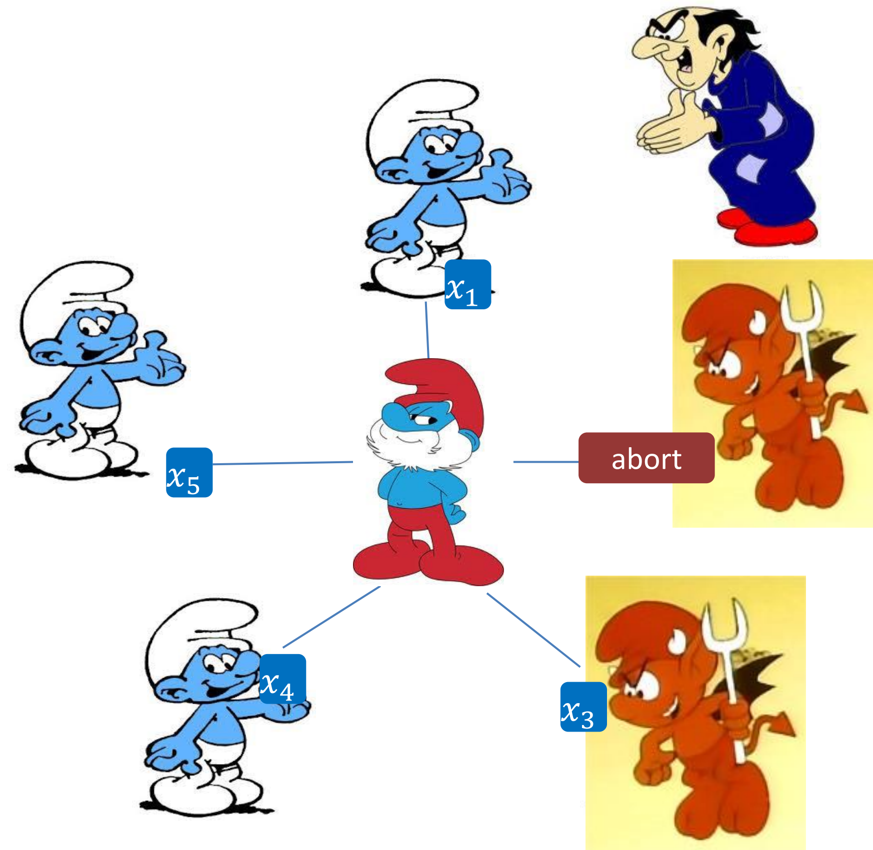
Security with G.O.D.

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



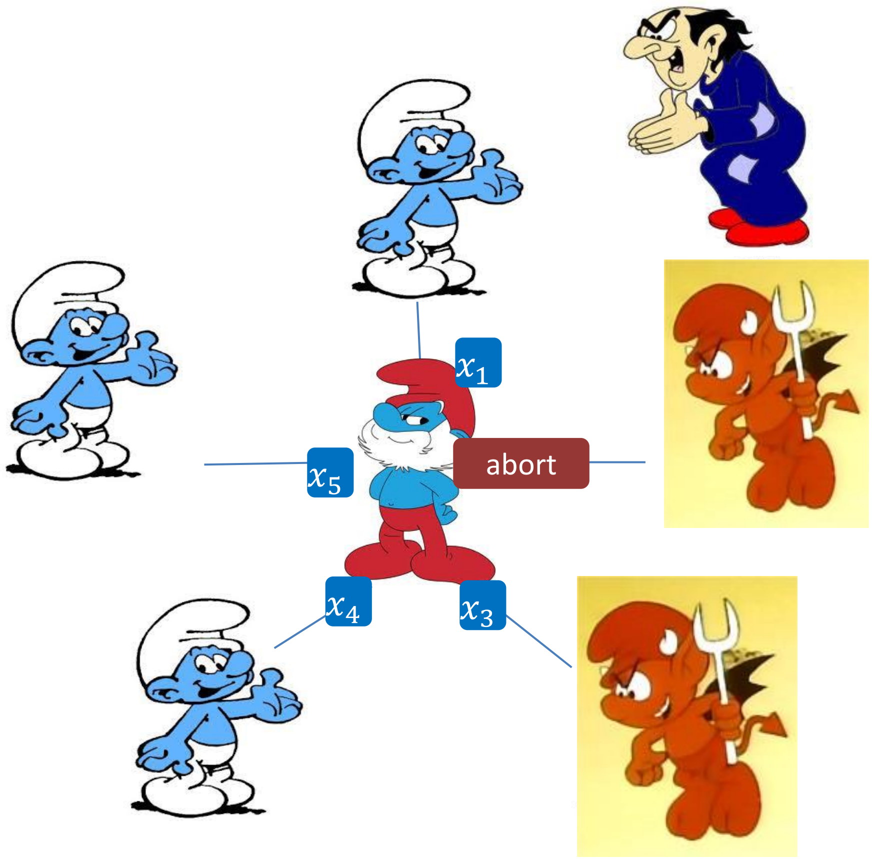
Security with fairness

1. Parties send input to $\mathcal{T}$



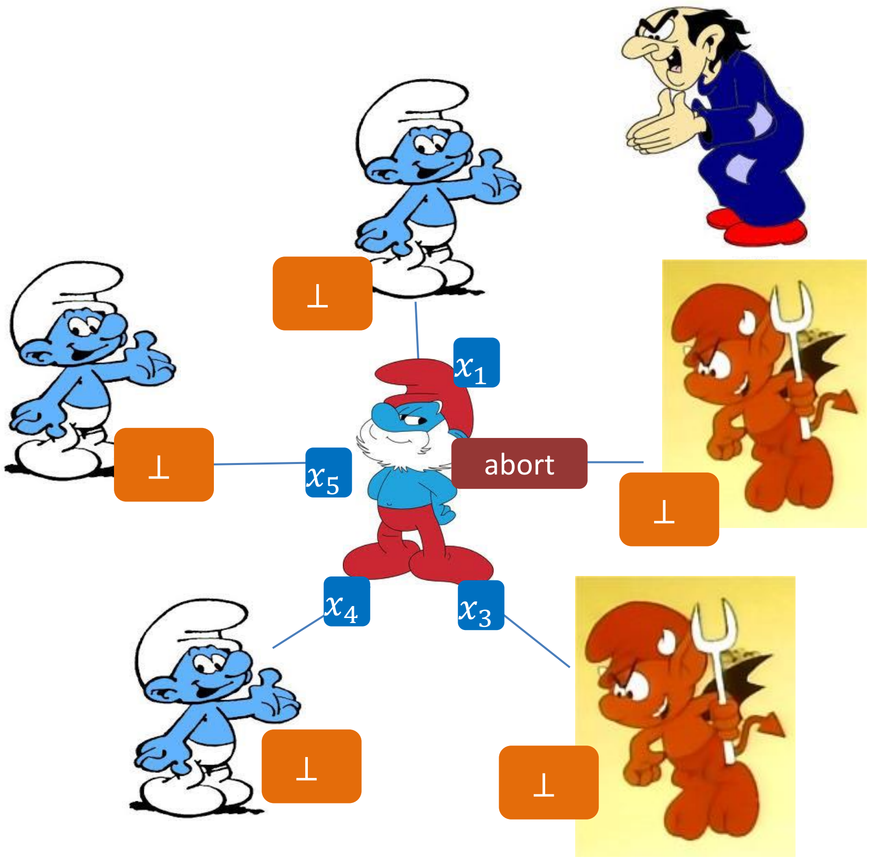
Security with fairness

1. Parties send input to $\mathcal{T}$
2. If $\mathcal{T}$ received $abort$, send $\perp$ to parties



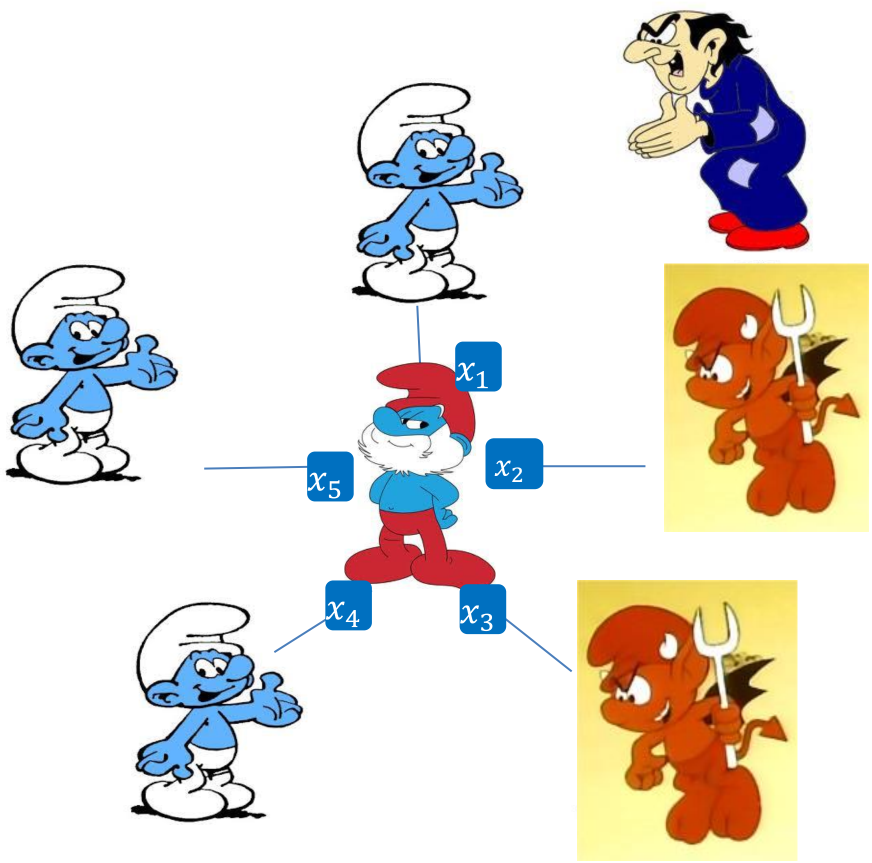
Security with fairness

1. Parties send input to $\mathcal{T}$
2. If $\mathcal{T}$ received $abort$, send $\perp$ to parties



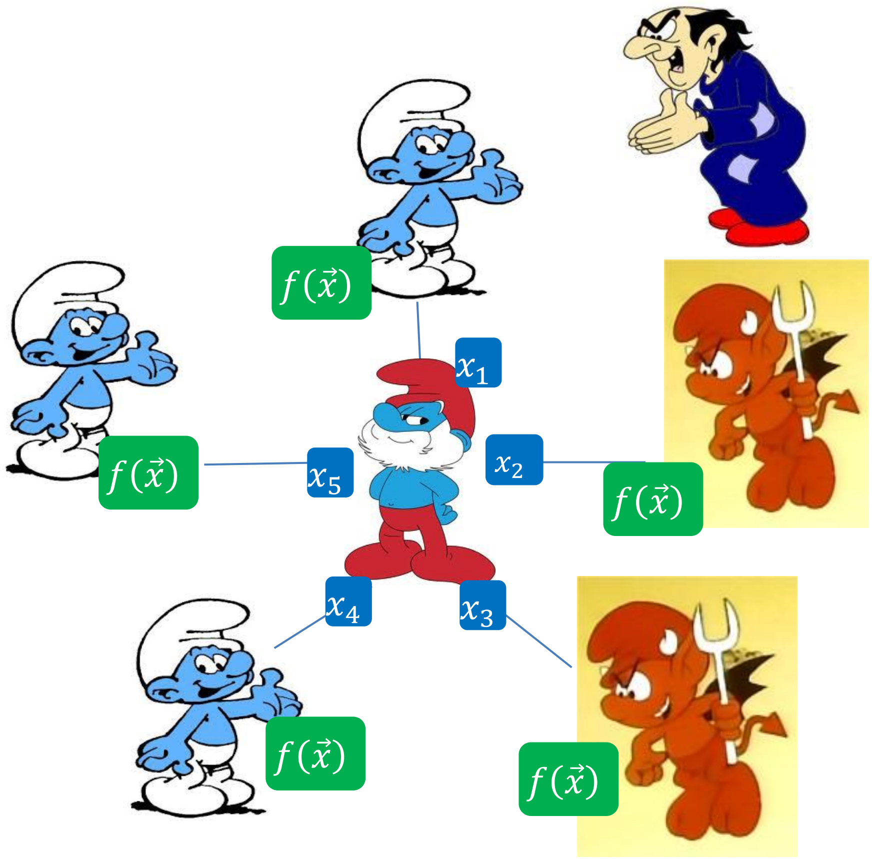
Security with fairness

1. Parties send input to $\mathcal{T}$
2. If $\mathcal{T}$ received $abort$, send $\perp$ to parties
3. Otherwise, $\mathcal{T}$ sends output to parties



Security with fairness

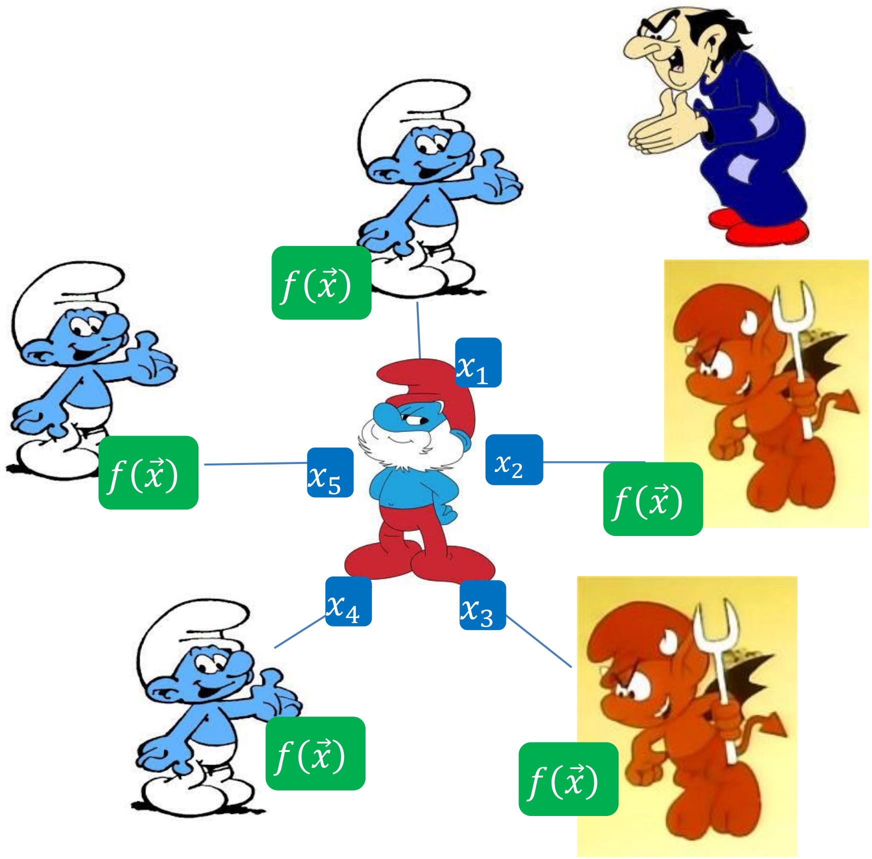
1. Parties send input to $\mathcal{T}$
2. If $\mathcal{T}$ received $abort$, send $\perp$ to parties
3. Otherwise, $\mathcal{T}$ sends output to parties



Security with fairness

1. Parties send input to $\mathcal{T}$
2. If $\mathcal{T}$ received $abort$, send $\perp$ to parties
3. Otherwise, $\mathcal{T}$ sends output to parties

Fairness with identifiable abort: $\mathcal{A}$ can send $(abort, i^*)$ and parties output $(\perp, i^*)$



Fairness $\nRightarrow$ G.O.D.

Fairness & broadcast

Lemma: fairness with broadcast $\Leftrightarrow$ fairness in P2P model

Proof:

- Let π be a fair protocol for f in the broadcast model
- Protocol with fairness for f in the P2P model:
  1) Compute PKI with abort as in [FGHHS'02]
  2) Run π with authenticated broadcast instead of broadcast
- Step (1) is independent of the inputs, so abort is fair
- Every abort in Step (2) is fair because π is fair

Separating fairness & G.O.D.

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

Idea: find a non-trivial f that

- Can be computed with fairness in P2P model
- Computing f with G.O.D. $\Rightarrow$ broadcast exists ($t \geq n/3$)
- No broadcast $\Rightarrow$ f cannot be computed with G.O.D.

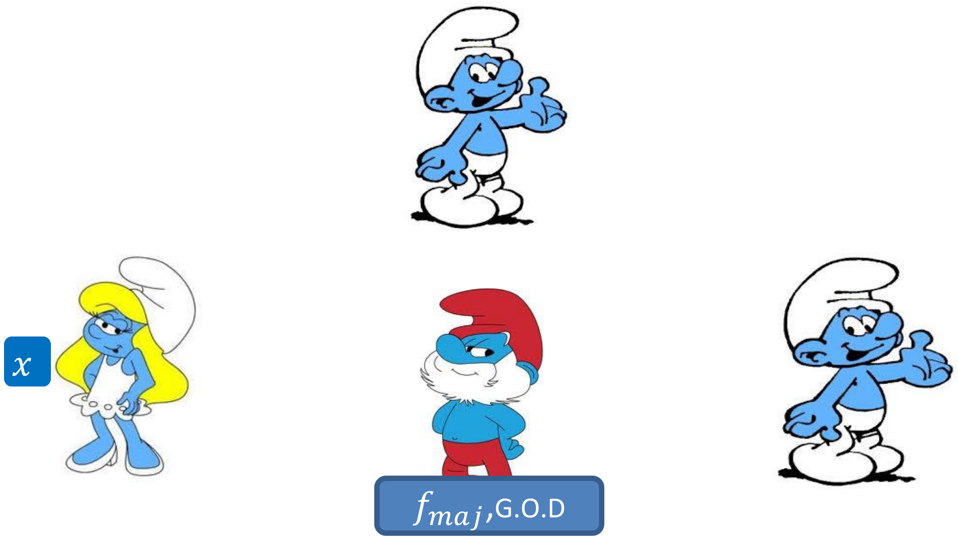
Three-party majority

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

- Fair with broadcast [GK'09] $\Rightarrow$ Fair in P2P model
- Non-trivial: 3-party f_{maj} $\Rightarrow$ 2-party OT [Kilian'91]

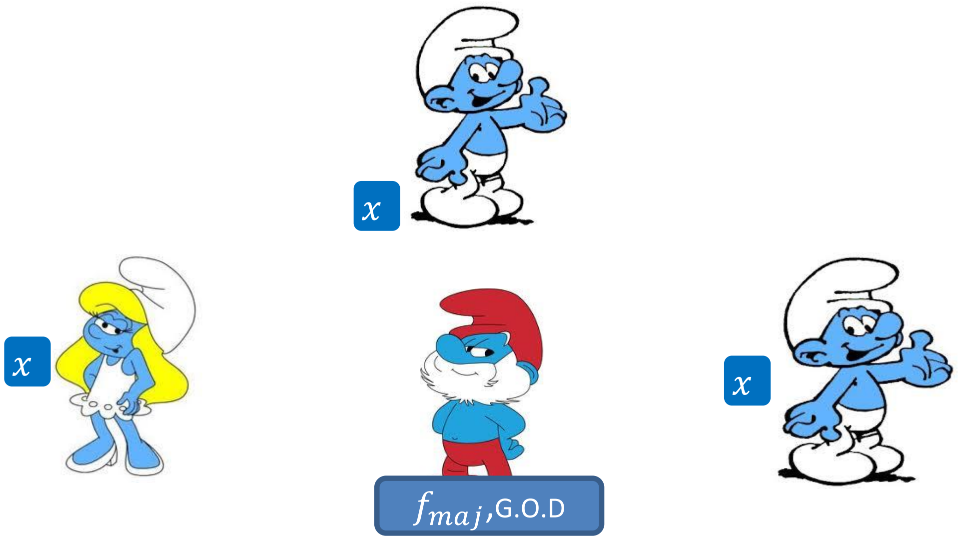
f_{maj} with G.O.D. $\Rightarrow$ broadcast

- Consider $\mathcal{T}$ that computes f_{maj} with G.O.D.
- Broadcast protocol in P2P model with $\mathcal{T}$:



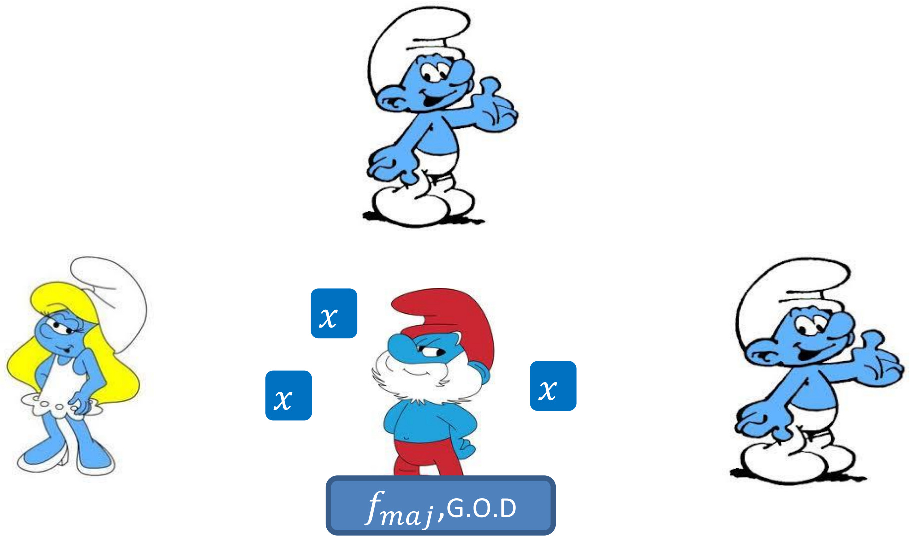
f_{maj} with G.O.D. ⇒ broadcast

- 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



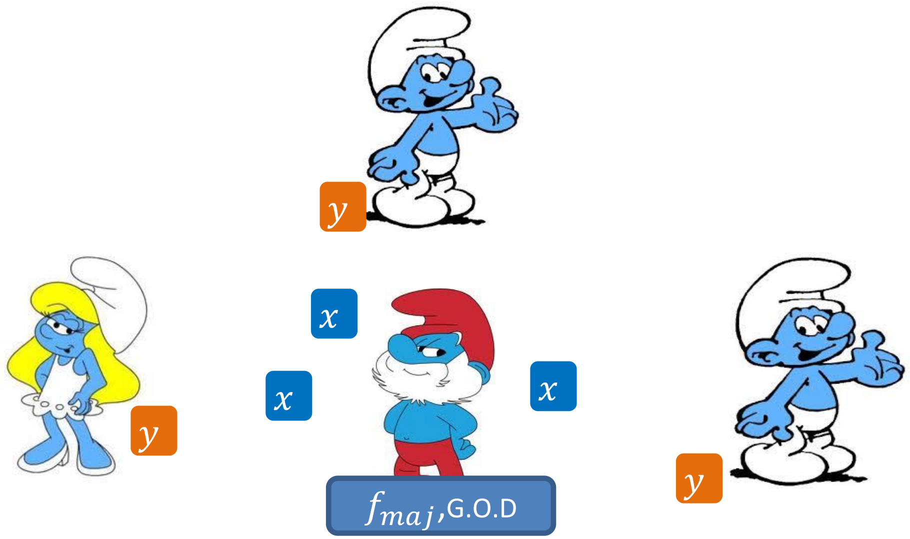
f_{maj} with G.O.D. ⇒ broadcast

- 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 $\mathcal{T}$



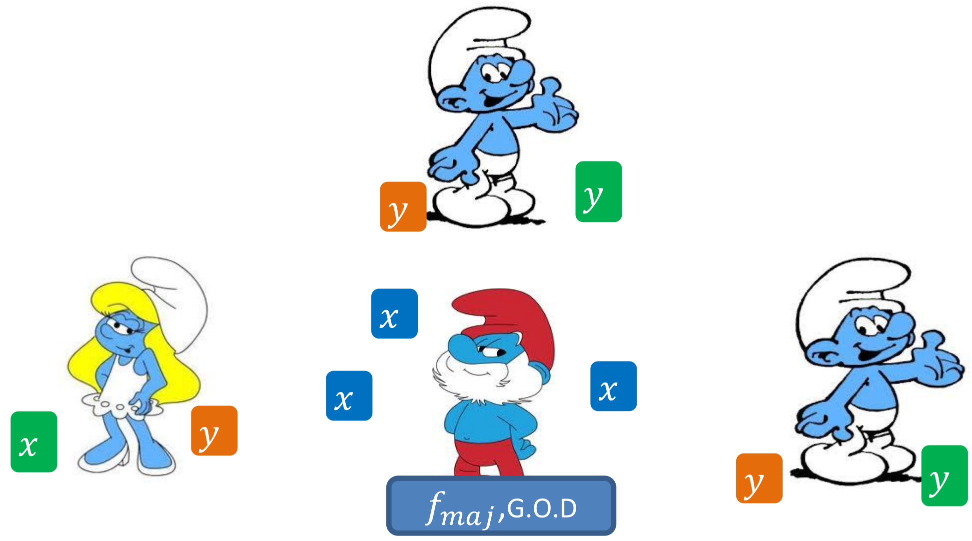
f_{maj} with G.O.D. ⇒ broadcast

- 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 $\mathcal{T}$



f_{maj} with G.O.D. ⇒ broadcast

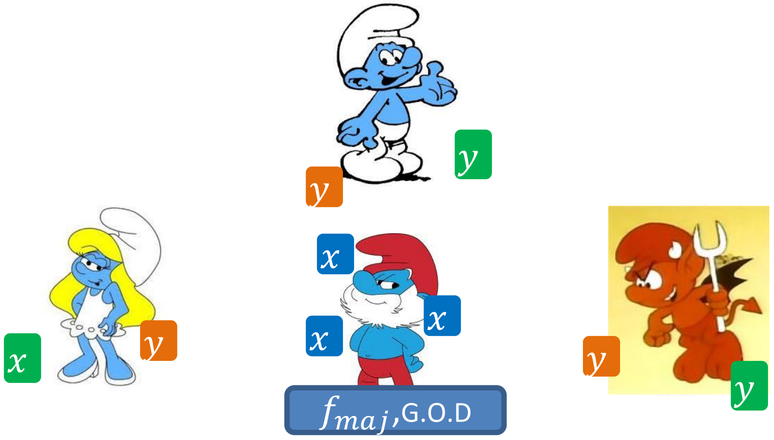
- Consider $\mathcal{T}$ that computes f_{maj} with G.O.D.
- Broadcast protocol in P2P model with $\mathcal{T}$:
  1. Sender sends $x \in \{0,1\}$ to all parties
  2. Each party sends its value to $\mathcal{T}$
  3. Each party gets $y \in \{0,1\}$ from $\mathcal{T}$
  4. Sender outputs x, receivers output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



f_{maj} with G.O.D. ⇒ broadcast

Intuition for the proof:

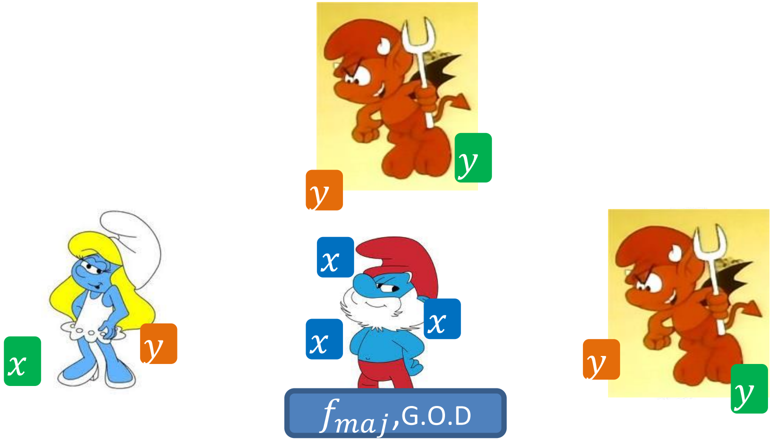
– Corrupted receiver: can send $\bar{x}$ to $\mathcal{T}$
  This doesn't affect the output of f_{maj}



f_{maj} with G.O.D. $\Rightarrow$ broadcast

Intuition for the proof:

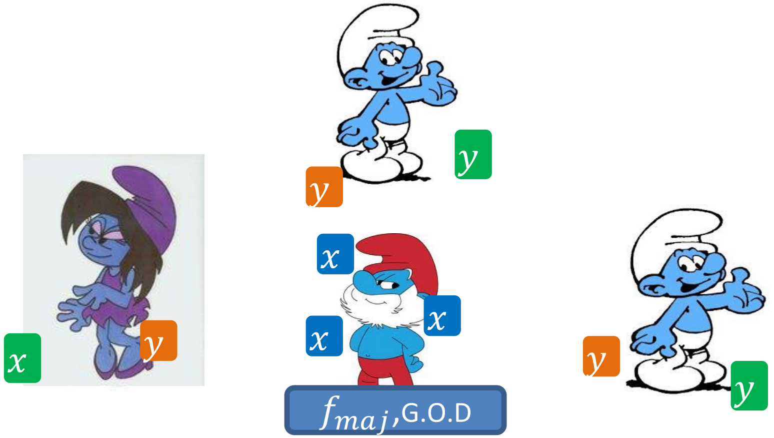
- Corrupted receiver: can send $\bar{x}$ to $\mathcal{T}$
  This doesn't affect the output of f_{maj}
- Two corrupted receivers: can determine the value y
  This doesn't affect the sender (always outputs x)



f_{maj} with G.O.D. $\Rightarrow$ broadcast

Intuition for the proof:

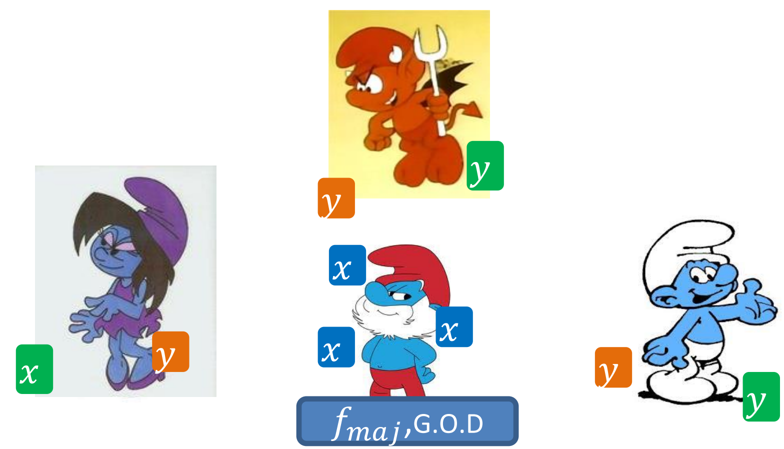
- Corrupted receiver: can send $\bar{x}$ to $\mathcal{T}$
  This doesn't affect the output of f_{maj}
- Two corrupted receivers: can determine the value y
  This doesn't affect the sender (always outputs x)
- Corrupted sender: can send different bits
  Both receivers obtain consist output y from $\mathcal{T}$



f_{maj} with G.O.D. ⇒ broadcast

Intuition for the proof:

- Corrupted receiver: can send $\bar{x}$ 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
  Both receivers obtain consist output y from $\mathcal{T}$
- Corrupted sender & receiver:
  No effect on honest receiver



Separating fairness & G.O.D.

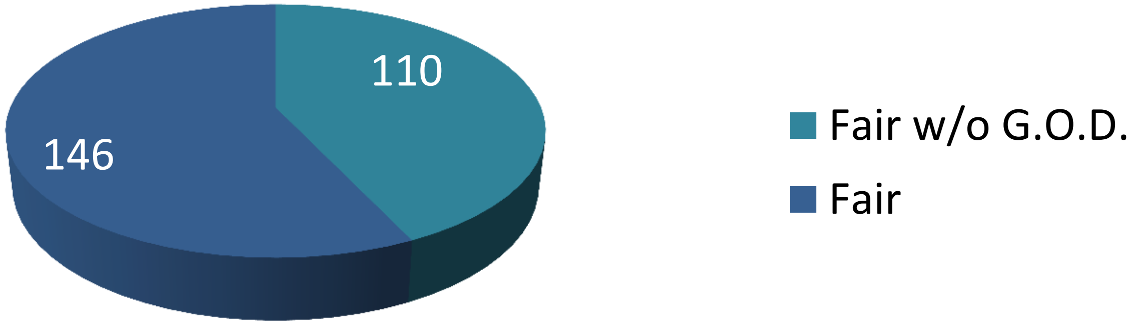
f_{maj} is fair without G.O.D. in P2P model $\forall t < 3$

We present a sufficient condition for f with G.O.D. $\Rightarrow$ broadcast

- Functions satisfying this condition are complete:

  If such f can be computed with G.O.D.,
  then every fair function can be computed with G.O.D.

- 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.

G.O.D. Without Broadcast

G.O.D. without broadcast

[GK'09] compute f_{maj} & f_{OR} in the broadcast model

f_{maj} cannot be computed with G.O.D. in the P2P model

Is broadcast needed for computing every f with G.O.D?

Multiparty Boolean OR

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

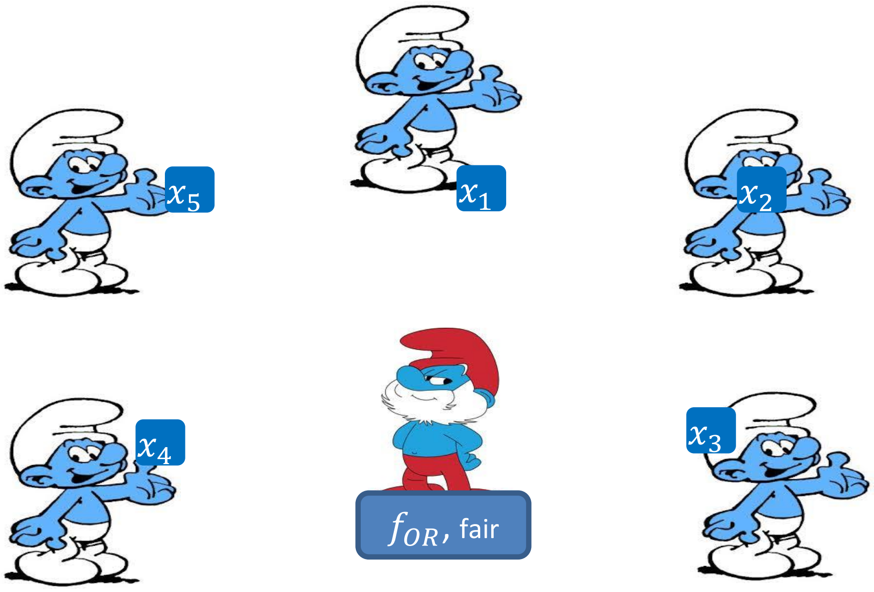
f_{OR} Can be computed with G.O.D. in the P2P model

Reason:

- Fair in P2P model (since fair in broadcast model)
- Every party can force the output to be 1

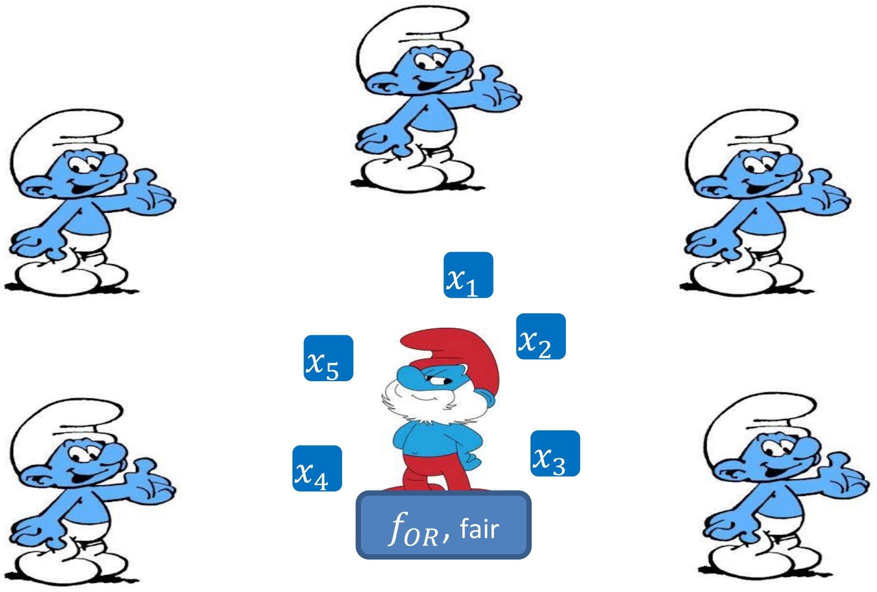
f_{OR} with G.O.D.

- Consider $\mathcal{T}$ that computes f_{OR} with fairness
- Protocol for f_{OR} with G.O.D. in P2P model & $\mathcal{T}$:



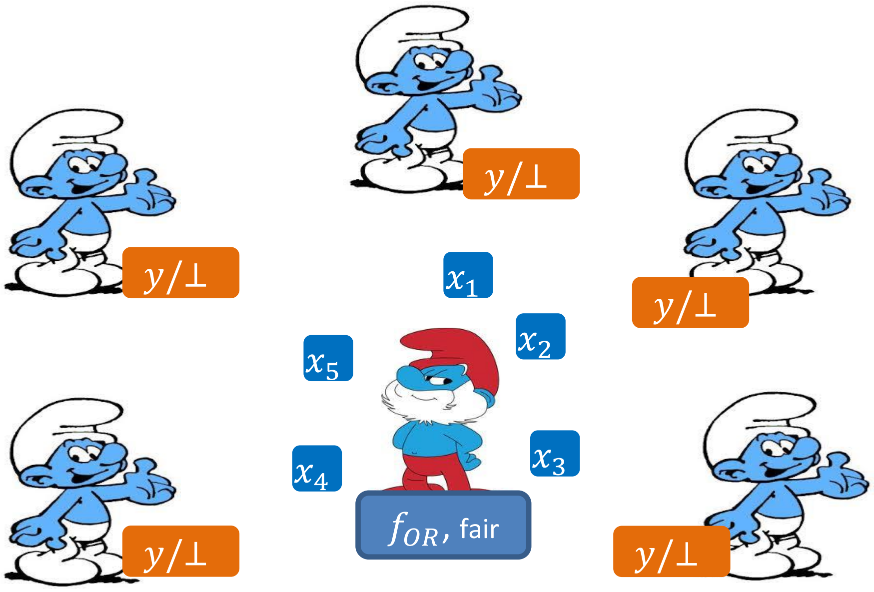
f_{OR} with G.O.D.

- 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 $x_i \in \{0,1\}$ to $\mathcal{T}$



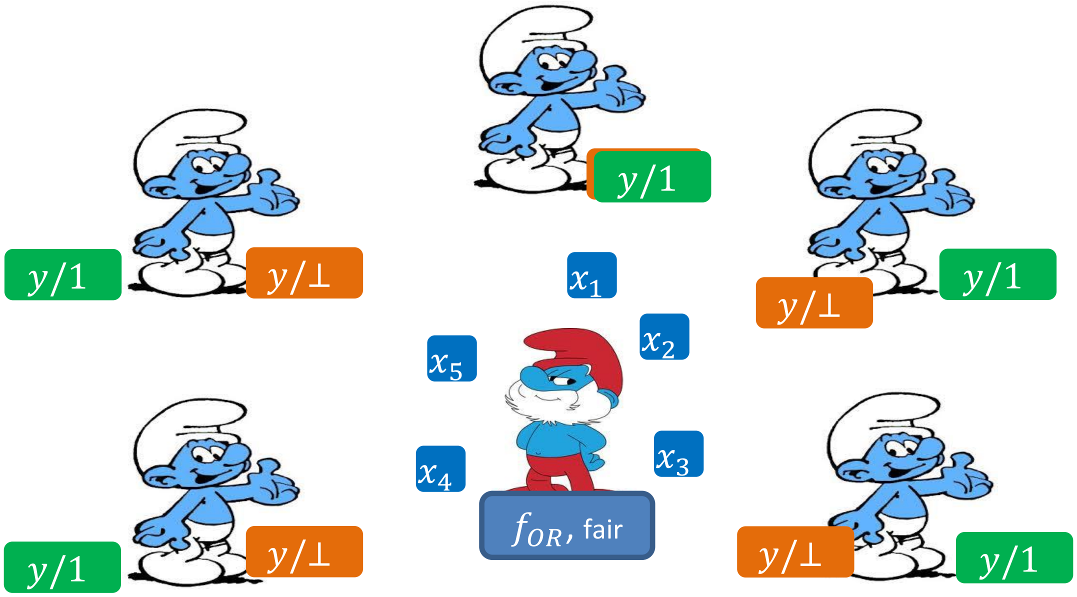
f_{OR} with G.O.D.

- 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 $x_i \in \{0,1\}$ to $\mathcal{T}$
  2. P_i receives $y/\perp$ from $\mathcal{T}$



f_{OR} with G.O.D.

- 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 $x_i \in \{0,1\}$ 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

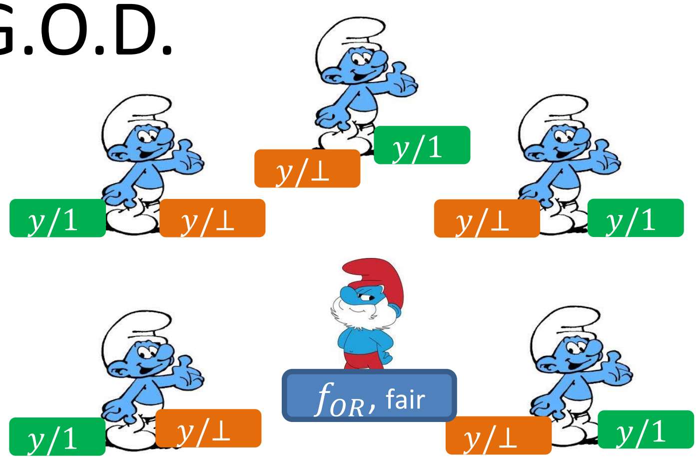
Intuition for the proof:

- 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 every party can force the output to be some default output value

f with this property is called 1-dominated

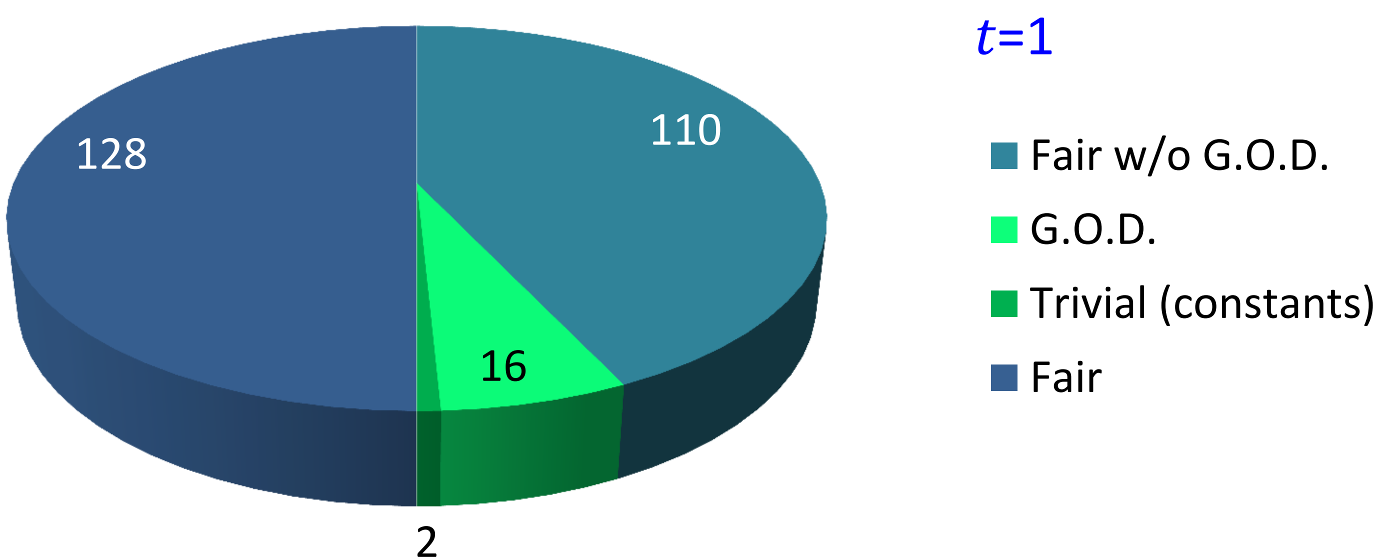
Cor: fairness & 1-dominated $\Rightarrow$ G.O.D.



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 and 1-dominated $\Rightarrow$ G.O.D. ($\forall t < 3$)



Conditions for fairness $\Rightarrow$ G.O.D.

Fairness & id-abort ⇒ G.O.D.

Recall Fairness & Identifiable Abort:

In case of a premature abort

- $\mathcal{A}$ 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

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)

2) If P_i fails to prove a message to P_j - the protocol resumes

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

When π completes:

- Either all parties learn the output
- Or all parties obtain $\perp$ and P_j identifies P_i as corrupted

➢ Fail-stop: P_j cannot falsely accuse P_i

Summary

- Fairness $\not\Leftrightarrow$ G.O.D. in P2P model
- Fairness $\Leftrightarrow$ G.O.D.
  - in the broadcast model
  - for 1-dominated functionalities
  - facing fail-stop adversaries

Thank You