Towards Characterizing Complete Fairness in Secure Two-Party Computation

Gilad Asharov

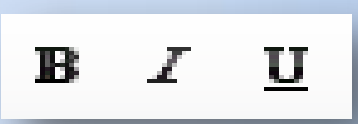
TCC 2014

Tradition of Excellence

Towards Characterizing Complete Fairness in Secure Two-Party Computation

Gilad Asharov

TCC 2014



Secure Multiparty Computation

n parties, each has some private input, wish to compute a function on their **joint** inputs

- average of salaries, auctions, private database query, private data mining

Secure Multiparty Computation

n parties, each has some private input, wish to compute a function on their **joint** inputs

- average of salaries, auctions, private database query, private data mining

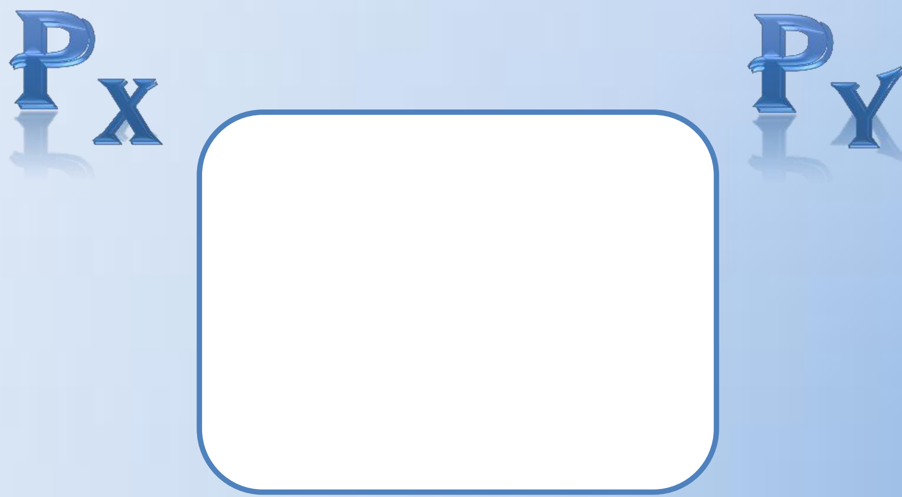
Security should be preserved even when some of the parties are **corrupted**

- correctness, privacy, independence of inputs and.. **fairness**

Complete Fairness

If the adversary learns the output, then all parties should learn also

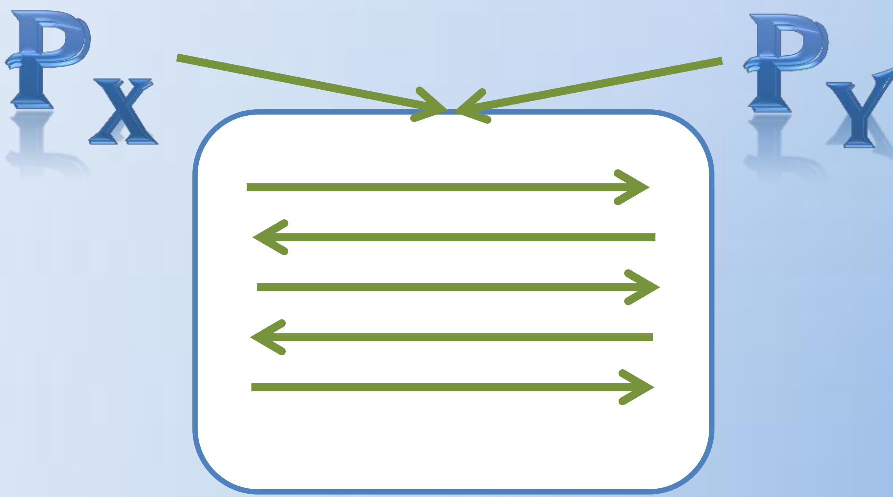
- In some sense, parties receive outputs simultaneously



Complete Fairness

If the adversary learns the output, then all parties should learn also

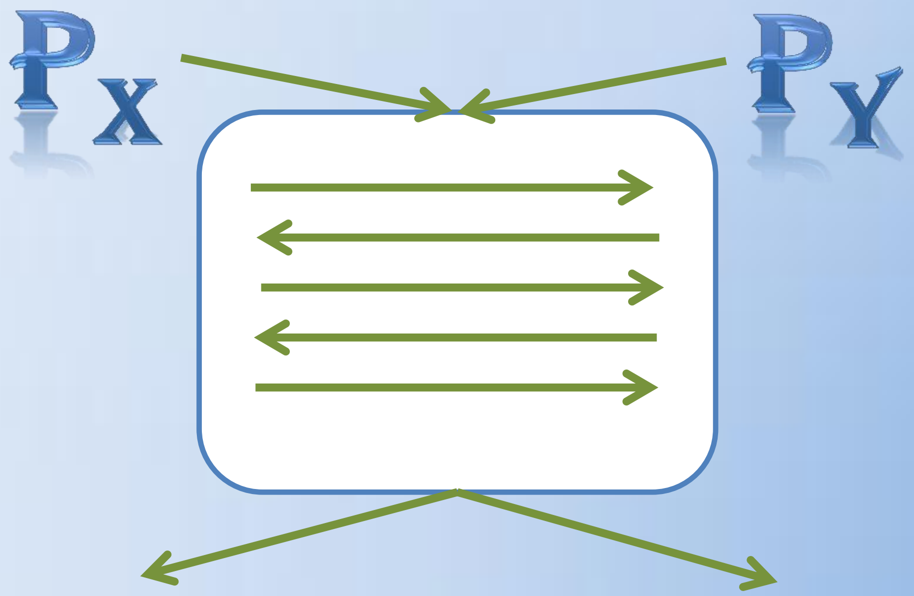
- In some sense, parties receive outputs simultaneously



Complete Fairness

If the adversary learns the output, then all parties should learn also

- In some sense, parties receive outputs simultaneously

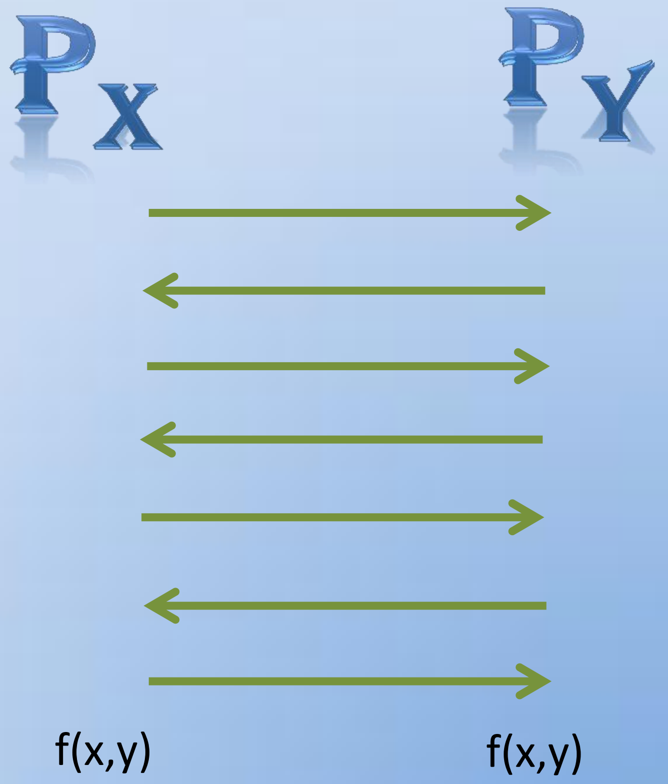


Complete Fairness

- **Complete fairness** can be achieved in multiparty with honest majority [GMW87,BGW88,CCD88,RB89,Be91]
- What about no honest majority?
  - Special case: *Two party setting*?

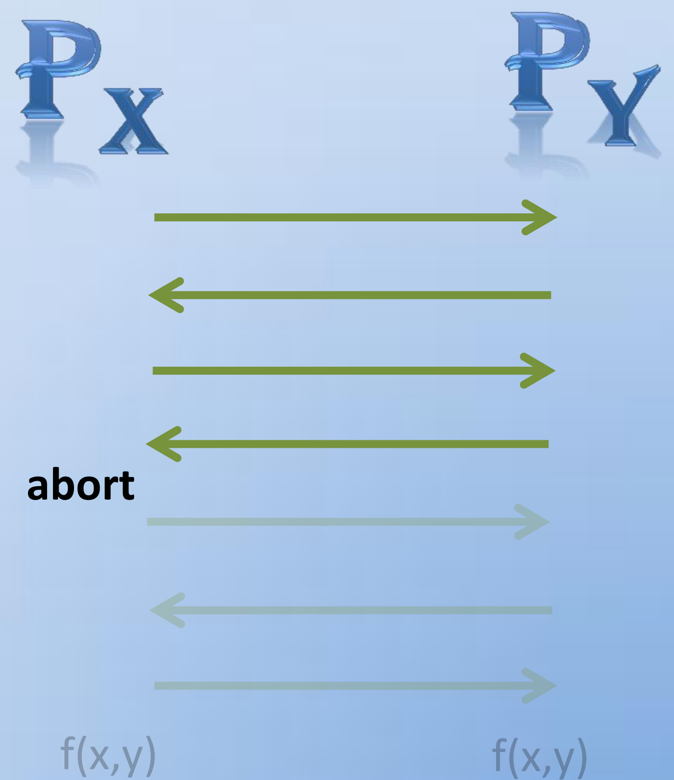
Difficulty of Fairness

- Beginning of execution – no knowledge about the outputs
- End of execution – full knowledge about it
- Protocols proceed in rounds
- The parties cannot exchange information simultaneously



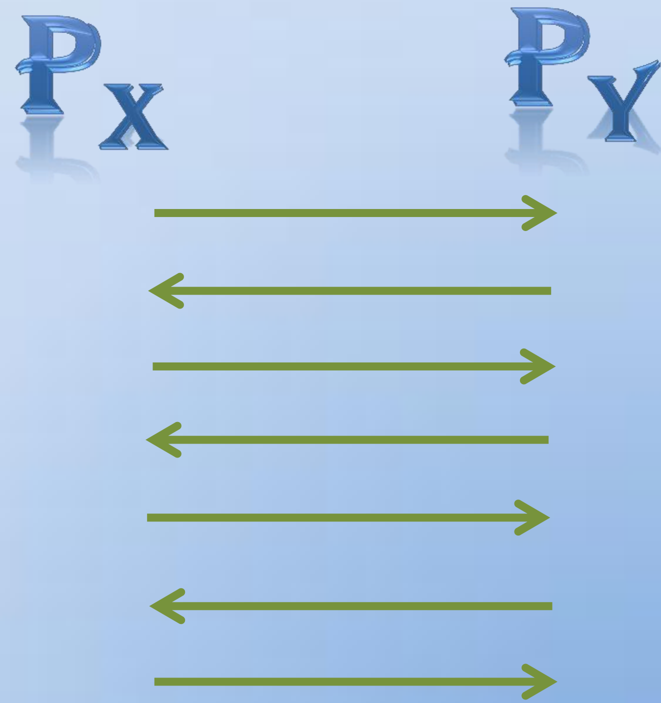
Difficulty of Fairness

- Beginning of execution – no knowledge about the outputs
- End of execution – full knowledge about it
- Protocols proceed in rounds
- The parties cannot exchange information simultaneously
- There must be a point when a party knows more than the other



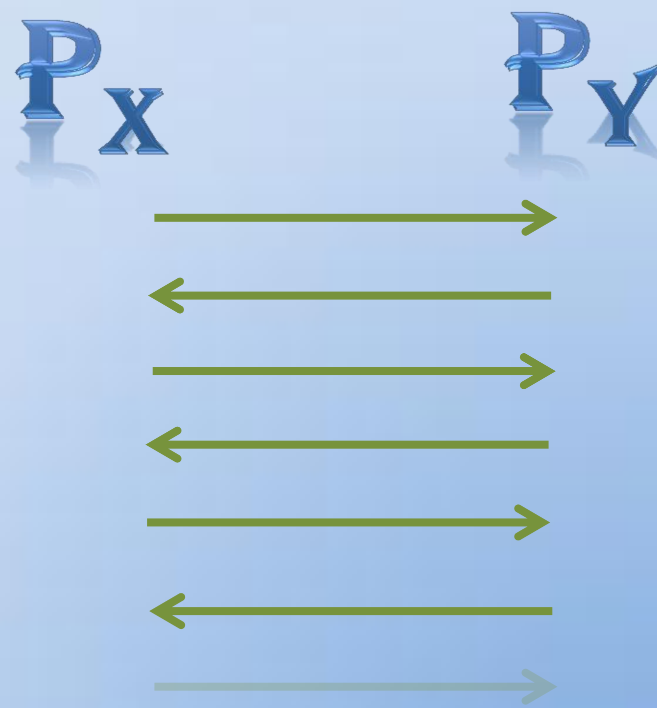
Difficulty of Fairness

- Take a fair protocol
- Remove the last round -> still fair protocol
- Continue the process..
- We stay with an empty protocol



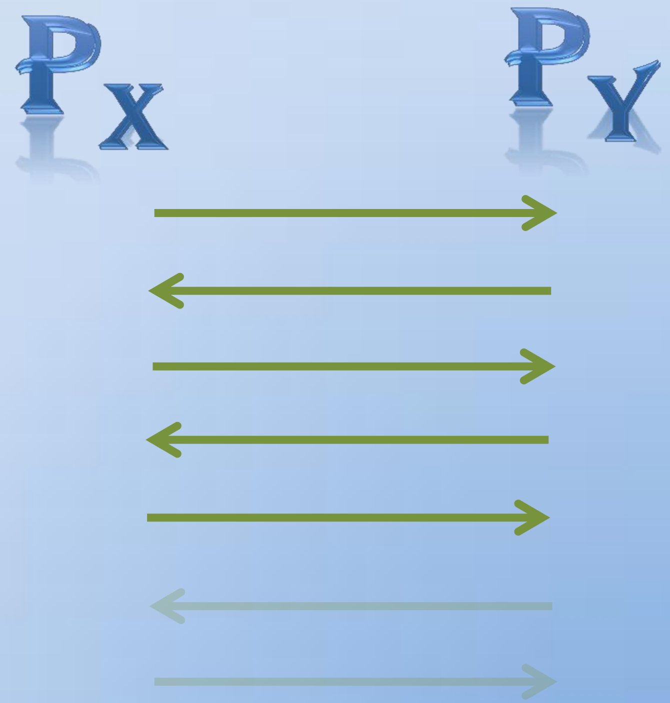
Difficulty of Fairness

- Take a fair protocol
- Remove the last round -> still fair protocol
- Continue the process..
- We stay with an empty protocol



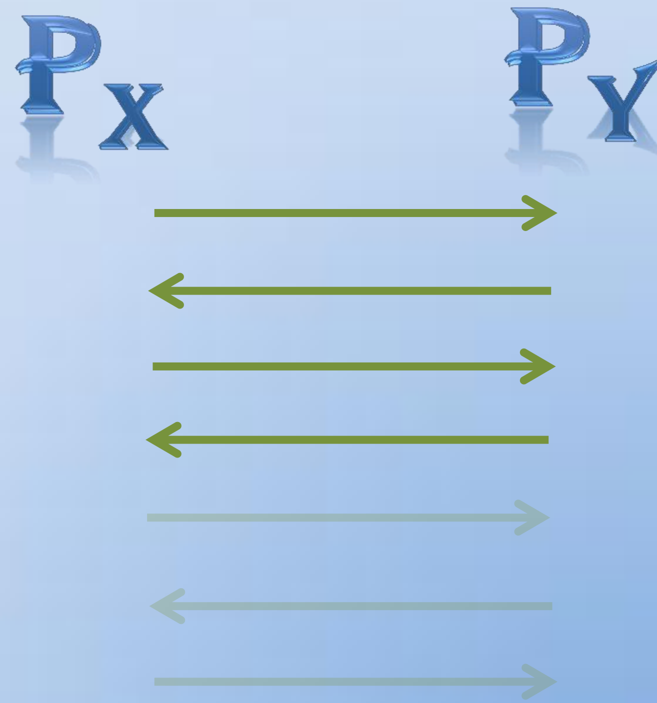
Difficulty of Fairness

- Take a fair protocol
- Remove the last round -> still fair protocol
- Continue the process..
- We stay with an empty protocol



Difficulty of Fairness

- Take a fair protocol
- Remove the last round -> still fair protocol
- Continue the process..
- We stay with an empty protocol



Complete Fairness

- In **1986**, Cleve showed that fairness is ***impossible*** in general (two party)

Complete Fairness

- In **1986**, Cleve showed that fairness is ***impossible*** in general (two party)
- The coin-tossing functionality is impossible:
  - both parties **agree** on the same uniform bit
  - **no** party can **bias** the result

Complete Fairness

- In **1986**, Cleve showed that fairness is ***impossible*** in general (two party)
- The coin-tossing functionality is impossible:
  - both parties **agree** on the same uniform bit
  - **no** party can **bias** the result
- Implies that the boolean XOR function is also impossible

	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**

Complete Fairness

- Since 1986, the accepted belief was that ***nothing*** non-trivial can be computed fairly

Complete Fairness

- Since 1986, the accepted belief was that ***nothing*** non-trivial can be computed fairly
- Many notions of partial fairness
  - Gradual release , Probabilistic fairness, Optimistic exchange, fairness at expectation [BeaverGoldwasser89][GoldwasserLevin90] [BonehNaor2000][Micali98]…

Complete Fairness

- Since 1986, the accepted belief was that ***nothing*** non-trivial can be computed fairly
- Many notions of partial fairness
  - Gradual release , Probabilistic fairness, Optimistic exchange, fairness at expectation [BeaverGoldwasser89][GoldwasserLevin90] [BonehNaor2000][Micali98]…
- Even two definitions of security – one with fairness, one without
- For two decades – no results on **complete fairness**

Complete Fairness

Gordon, Hazay, Katz and Lindell [STOC08] showed that there exist some **non-trivial** functions that can be computed with **complete fairness**!

Complete Fairness

Gordon, Hazay, Katz and Lindell [STOC08] showed that there exist some **non-trivial** functions that can be computed with **complete fairness**!

	y_1	y_2	y_3	y_4	y_5
x_1	0	0	0	0	0
x_2	1	0	0	0	0
x_3	1	1	0	0	0
x_4	1	1	1	0	0
x_5	1	1	1	1	0

Complete Fairness

Gordon, Hazay, Katz and Lindell [STOC08] showed that there exist **some** **non-trivial** functions that can be computed with **complete fairness**!

	y_1	y_2	y_3	y_4	y_5
x_1	0	0	0	0	0
x_2	1	0	0	0	0
x_3	1	1	0	0	0
x_4	1	1	1	0	0
x_5	1	1	1	1	0

	y_1	y_2
x_1	0	1
x_2	1	0
x_3	1	1

Characterizing Fairness

- A fundamental question:

What functions can and cannot be securely computed with complete fairness?

Characterizing Fairness

- **A fundamental question:**

What functions can and cannot be securely computed with complete fairness?

- Impossibility: Cleve

Characterizing Fairness

- **A fundamental question:**

What functions can and cannot be securely computed with complete fairness?

- Impossibility: Cleve
- Only few examples of functions that are possible

Two Works

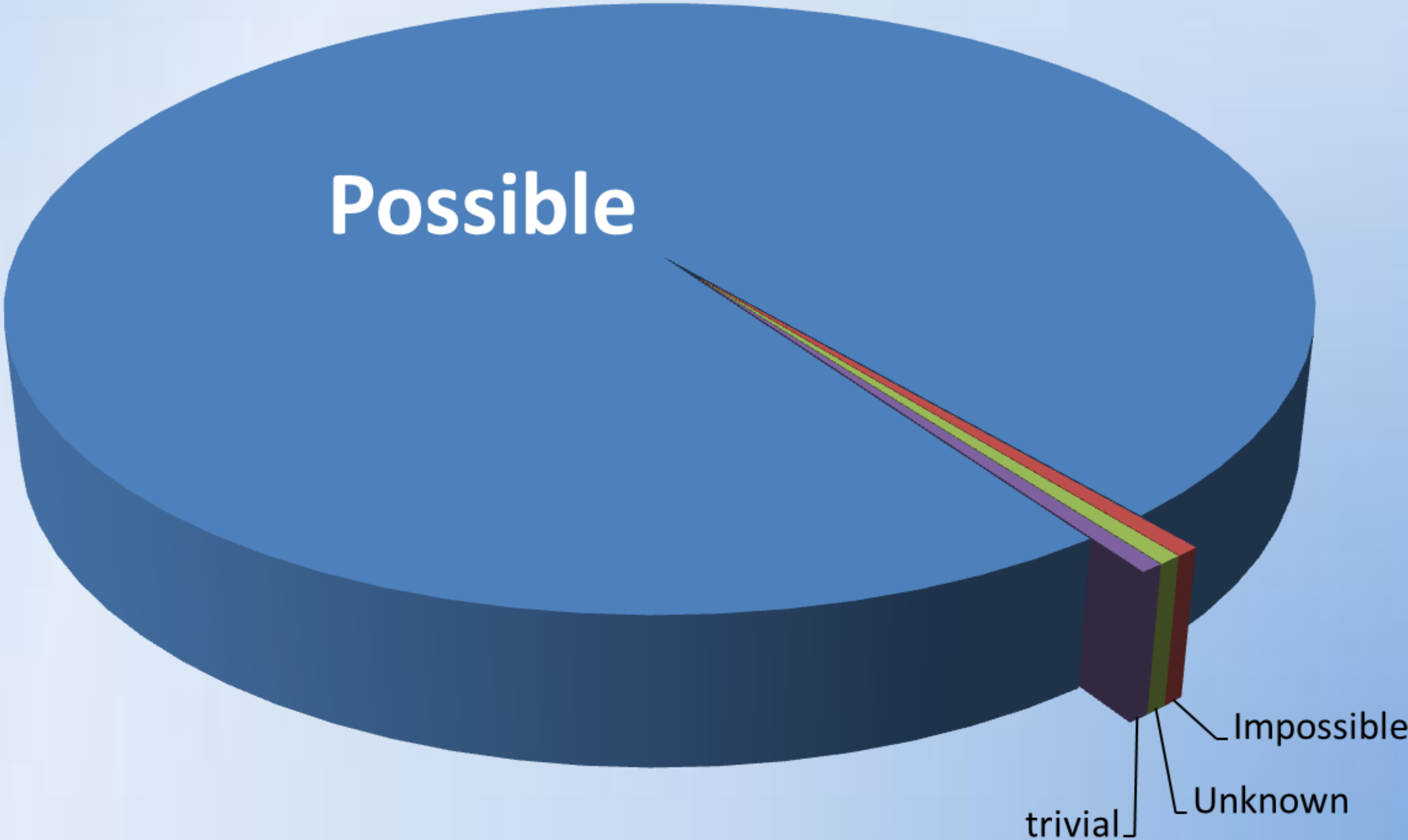
- **A Full Characterization of Functions that Imply Fair Coin Tossing and Ramifications to Fairness**
  A, Lindell and Rabin [TCC 2013]

- **Towards Characterizing Complete Fairness in Secure Two-Party Computing**
  A [TCC 2014]

$$f: X \times Y \longrightarrow \{0,1\}$$

with $|X| \neq |Y|$

$f: X \times Y \longrightarrow \{0,1\}$
with $|X| \neq |Y|$



Examples

Set Membership

- **X input:** $S \subseteq \Omega$ (possible inputs: $2^{|\Omega|}$)
- **Y input:** $\omega \in \Omega$ (possible inputs: $|\Omega|$)
- The function $f(S, \omega) = \omega \in S$?

Examples

Set Membership

- **X input:** $S \subseteq \Omega$ (possible inputs: $2^{|\Omega|}$)
- **Y input:** $\omega \in \Omega$ (possible inputs: $|\Omega|$)
- The function $f(S, \omega) = \omega \in S$?

Private Evaluation of a Boolean Function

- **X input:** $g \in \mathrm{F}$ $(F = \{g: \Omega \to \{0,1\}\})$
- **Y input:** $y \in \Omega$
- The function $f(g, y) = g(y)$

Examples

Private Matchmaking:

- X holds set of preferences (“what I am looking for”)
- Y holds a profile (“who I am”)
- Output: Does Y match X

Examples

Private Matchmaking:

- X holds set of preferences ("what I am looking for")
- Y holds a profile ("who I am")
- Output: Does Y match X

$\boldsymbol{A \subseteq B}$:

- X holds $A \subseteq \Omega$
- Y holds $B \subseteq \Omega$
- Output: $A \subseteq B$?

Examples

Private Matchmaking:

- X holds set of preferences ("what I am looking for")
- Y holds a profile ("who I am")
- Output: Does Y match X

$\boldsymbol{A \subseteq B}$:

- X holds $A \subseteq \Omega$
- Y holds $B \subseteq \Omega$
- Output: $A \subseteq B$?

Set Disjointness:

- X holds $A \subseteq \Omega$
- Y holds $B \subseteq \Omega$
- Output: $A \cap B = \emptyset$?

Examples

$$\begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 1 & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Examples

$$\begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 1 & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Impossible

$A = B$

implies coin-tossing

[ALR13]

Examples

$$\begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 1 & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Impossible

$A = B$

implies coin-tossing

[ALR13]

Possible

$A \subseteq B$

Examples

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Impossible

$A = B$

implies coin-tossing
[ALR13]

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Unknown

not coin-tossing
not [GHKL08]*

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Possible

$A \subseteq B$

A Full Characterization of Functions that Imply Fair Coin Tossing and Ramifications to Fairness

Asharov, Lindell, Rabin

TCC 2013

Coin-Tossing Impossibility [Cleve86]

The coin-tossing functionality is impossible:

$$f(\lambda, \lambda) = (U, U)$$

(U is the uniform distribution over {0,1})

– both parties **agree** on the same uniform bit

– **no** party can **bias** the result

Coin-Tossing Impossibility [Cleve86]

The coin-tossing functionality is impossible:

$$f(\lambda,\lambda) = (U,U)$$

(U is the uniform distribution over {0,1})

– both parties **agree** on the same uniform bit

– **no** party can **bias** the result

Question:

Which Boolean functions are ruled out by this impossibility?

Which functions imply fair coin-tossing?

The XOR Function

	y_1	y_2
x_1	0	1
x_2	1	0

Question:

Assume a fair protocol for the XOR function

How can we use it to toss a coin?

The XOR Function

	y_1	y_2
x_1	0	1
x_2	1	0

Question:

Assume a fair protocol for the XOR function

How can we use it to toss a coin?

Answer:

Each party chooses a uniform bit, then XOR them

Why Does it Work?

$$\Pr[output = 1] = \underbrace{\begin{pmatrix} p_1 & p_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{X}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \underbrace{\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{Y}}$$

Why Does it Work?

$$\Pr[output = 1] = \underbrace{\begin{pmatrix} p_1 & p_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{X}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \underbrace{\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{Y}}$$

distribution over the inputs of **X**

distribution over the inputs of **Y**

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Why Does it Work?

$$\Pr[output = 1] = \begin{pmatrix} p_1 & p_2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

distribution over the inputs of **X**

distribution over the inputs of **Y**

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \frac{1}{2}$$

Why Does it Work?

$$\Pr[output = 1] = \underbrace{\begin{pmatrix} p_1 & p_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{X}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \underbrace{\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{Y}}$$

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \frac{1}{2}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$

Why Does it Work?

$$\Pr[output = 1] = \underbrace{\begin{pmatrix} p_1 & p_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{X}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \underbrace{\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}}_{\text{distribution over the inputs of } \mathbf{Y}}$$

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \frac{1}{2}$$

$$\begin{pmatrix} p_1 & p_2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} p_1 & p_2 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \frac{1}{2}$$

The Property

f is δ balanced

if there exist probability vectors $\boldsymbol{p} = (p_1, \ldots, p_m)$,
$\boldsymbol{q} = (q_1, \ldots, q_\ell)$ and $0 < \delta < 1$ s.t:

$\boldsymbol{p} \cdot M_f = \delta \cdot \mathbf{1}_\ell$ **AND** $M_f \cdot \boldsymbol{q}^T = \delta \cdot \mathbf{1}_m^T$

The Property

f is δ balanced

if there exist probability vectors $\boldsymbol{p} = (p_1, \dots, p_m)$, $\boldsymbol{q} = (q_1, \dots, q_\ell)$ and $0 < \delta < 1$ s.t:

$$\boldsymbol{p} \cdot M_f = \delta \cdot \mathbf{1}_\ell \quad \textbf{AND} \quad M_f \cdot \boldsymbol{q}^T = \delta \cdot \mathbf{1}_m^T$$

Theorem

If f is δ-balanced then it implies fair coin-tossing

Other Examples

Balanced Functions:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Unbalanced Functions:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

(left-balanced, right-unbalanced)

Other Examples

Balanced Functions:

$$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} \qquad \begin{pmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{pmatrix} \qquad \begin{pmatrix}1 & 0 & 1\\0 & 1 & 0\\0 & 0 & 1\end{pmatrix}$$

Unbalanced Functions:

$$\begin{pmatrix}1 & 0\\0 & 1\\1 & 1\end{pmatrix} \qquad \begin{pmatrix}1 & 0\\0 & 1\\1 & 1\end{pmatrix}\begin{pmatrix}p\\1-p\end{pmatrix}=\begin{pmatrix}p\\1-p\\1\end{pmatrix}$$

(left-balanced, right-unbalanced)

This is Tight!*

Theorem

if f is not δ-balanced for any $0 < \delta < 1$, then it **does not imply** coin tossing*

This is Tight!*

Theorem

if f is not δ-balanced for any $0 < \delta < 1$, then it **does not imply** coin tossing*

- We show that for any coin-tossing protocol in the f-hybrid model, there exists an adversary that can bias the result

This is Tight!*

Theorem

if f is not δ-balanced for any $0 < \delta < 1$, then it **does not imply** coin tossing*

- We show that for any coin-tossing protocol in the f-hybrid model, there exists an adversary that can bias the result
- Unlike Cleve – here we do have something simultaneously. A completely different argument is given

This is Tight!*

Theorem

if f is not δ-balanced for any $0 < \delta < 1$, then it **does not imply** coin tossing*

- We show that for any coin-tossing protocol in the f-hybrid model, there exists an adversary that can bias the result
- Unlike Cleve – here we do have something simultaneously. A completely different argument is given
- **Caveat**: the adversary is **inefficient**

This is Tight!*

Theorem

if f is not δ-balanced for any $0 < \delta < 1$, then it **does not imply** coin tossing*

- We show that for any coin-tossing protocol in the f-hybrid model, there exists an adversary that can bias the result
- Unlike Cleve – here we do have something simultaneously. A completely different argument is given
- **Caveat**: the adversary is **inefficient**
- However, impossibility holds also when the parties have OT-oracle (and so commitments, ZK, etc.)

Towards Characterizing Complete Fairness in Secure Two-Party Computation

Asharov

TCC 2014

The Protocol of [GHKL08]

Gordon, Hazay, Katz and Lindell [STOC08] presented a general protocol and proved that a particular function can be computed using this protocol

	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**
x_3	**1**	**1**

The Protocol of [GHKL08]

Gordon, Hazay, Katz and Lindell [STOC08] presented a general protocol and proved that a particular function can be computed using this protocol

	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**
x_3	**1**	**1**

Question:

What functions can be computed using this protocol?

The Result

- **Almost all functions with $|\mathbf{X}| \neq |\mathbf{Y}|$:**
  can be computed using the protocol
- **Almost all functions with $|\mathbf{X}| = |\mathbf{Y}|$:**
  cannot be computed using the protocol
  - If the function has monochromatic input, it may be possible even if $|X| = |Y|$
- **Characterization of [GHKL08] is not tight!**
  - There are functions that are left unknown

The Protocol of [GHKL08]

- Special round i^*
- Until round i^* - the outputs are random and uncorrelated $(f(x,\hat{y}), f(\hat{x},y))$
- Starting at i^* - the outputs are correct
- At i^*, P_x learns before P_y

The Protocol of [GHKL08]

- Special round i^*
- Until round i^* - the outputs are random and uncorrelated $(f(x, \hat{y}), f(\hat{x}, y))$
- Starting at i^* - the outputs are correct
- At i^*, P_x learns before P_y
- Security:
  - P_y is always the **second** to receive output
    - Simulation is possible for **all** functions
  - P_x is always the **first** to receive output
    - Simulation is possible only for **some** functions

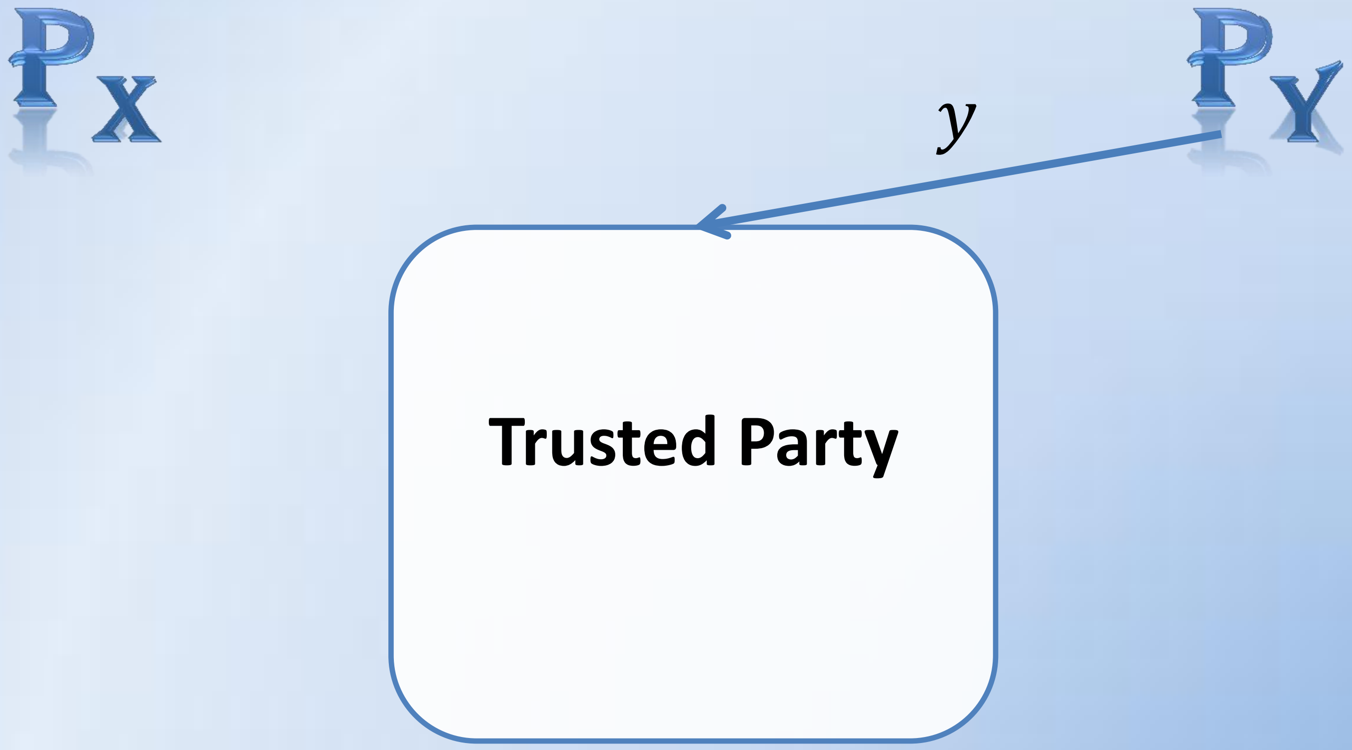
The Definition

P_X

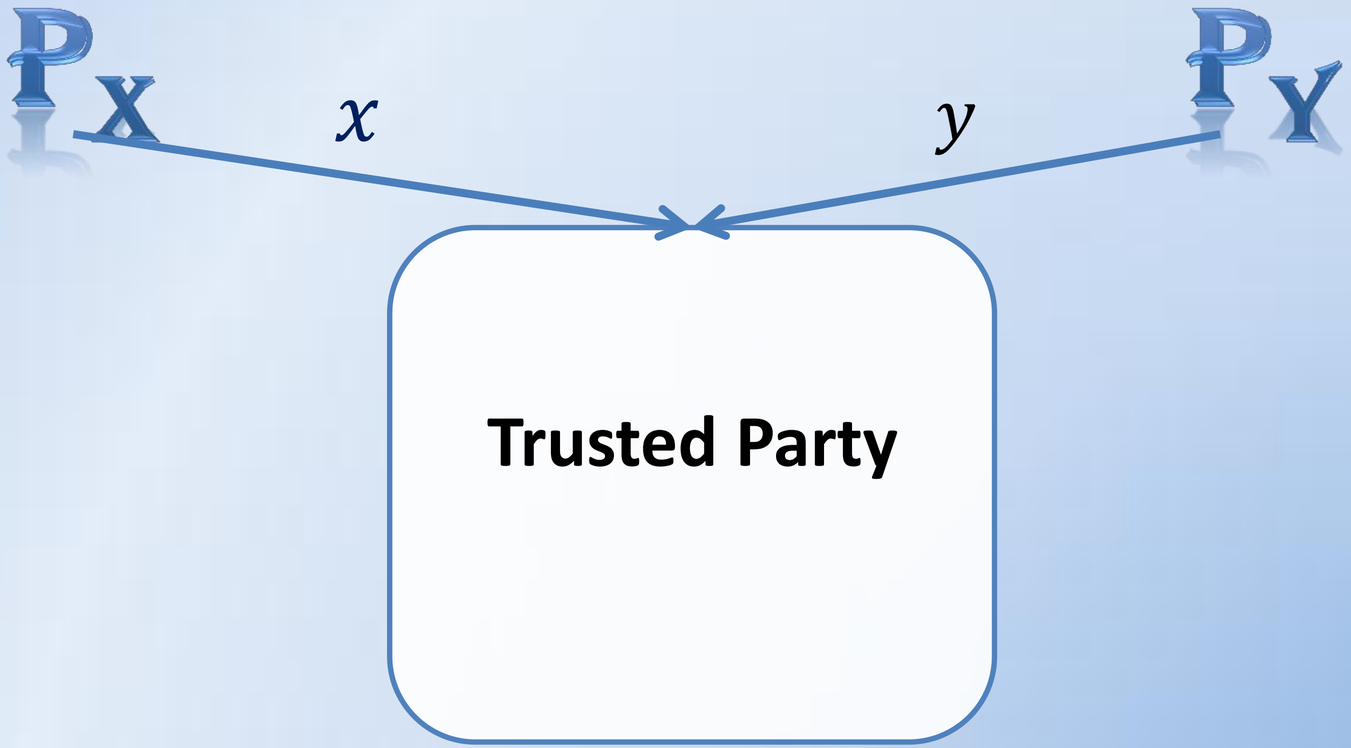
P_Y

Trusted Party

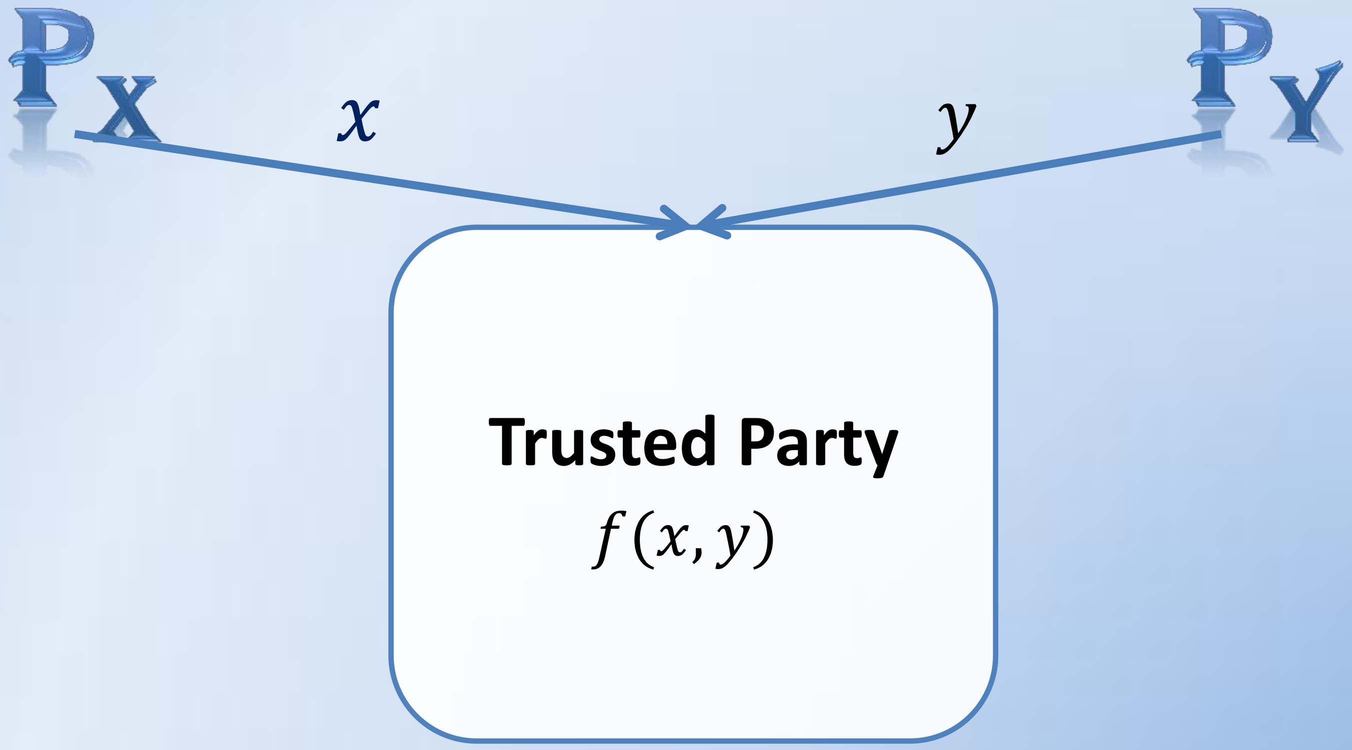
The Definition



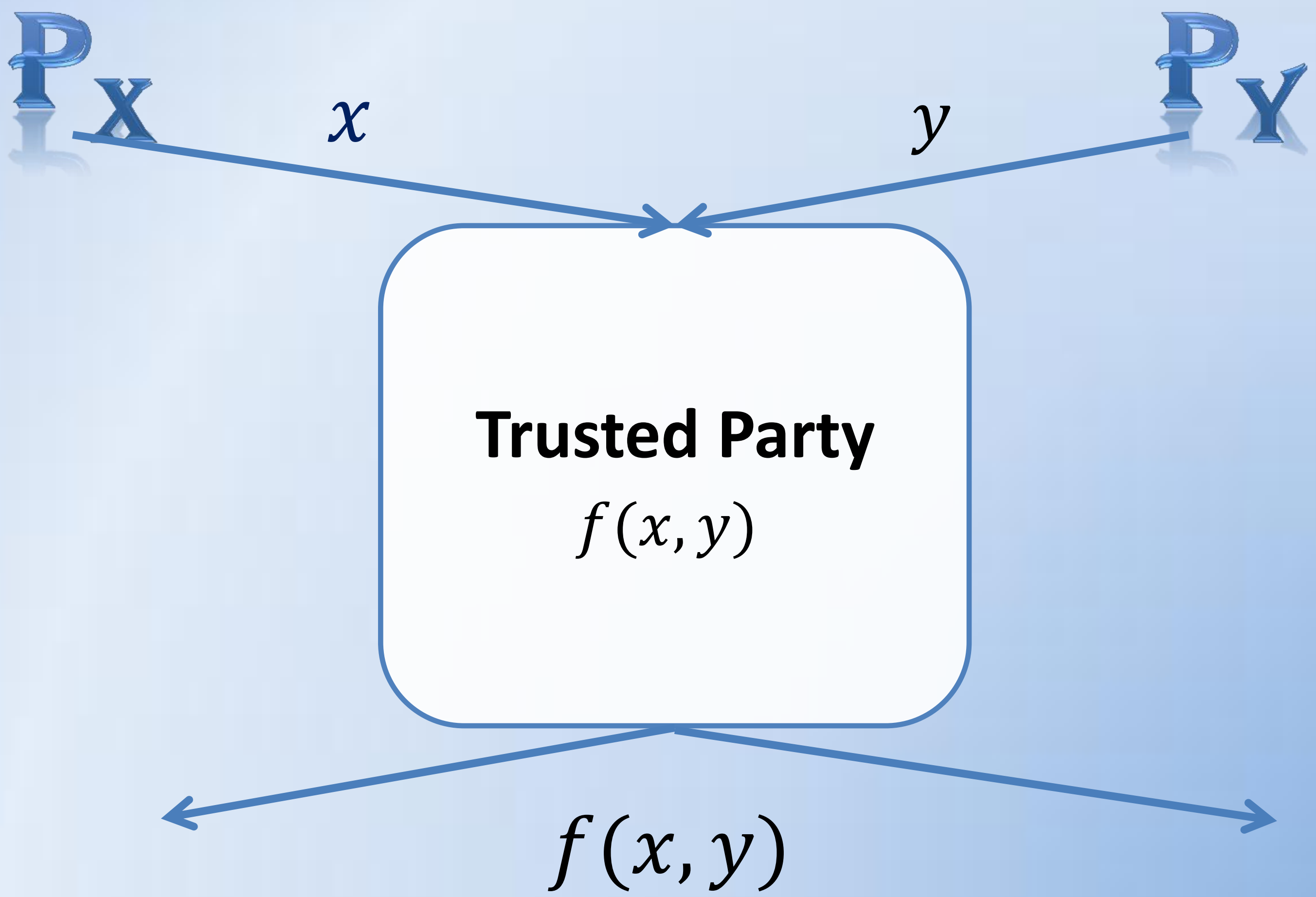
The Definition



The Definition



The Definition



Manipulating Output (Possible)

Before i^* : $f(\hat{x}, y)$

		y_1	y_2
1/3	x_1	**0**	**1**
1/3	x_2	**1**	**0**
1/3	x_3	**1**	**1**

$$\left(\frac{2}{3}, \frac{2}{3}\right)$$

Manipulating Output (Possible)

Before i^* : $f(\hat{x}, y)$

		y_1	y_2
1/3	x_1	**0**	**1**
1/3	x_2	**1**	**0**
1/3	x_3	**1**	**1**

$$\left(\frac{2}{3}, \frac{2}{3}\right)$$

$$\left(\frac{2}{3} + \epsilon, \frac{2}{3}\right)$$

Manipulating Output (Possible)

Before i^* : $f(\hat{x}, y)$

			y_1	y_2
$1/3-\epsilon$	1/3	x_1	**0**	**1**
$1/3$	1/3	x_2	**1**	**0**
$1/3+\epsilon$	1/3	x_3	**1**	**1**

$$(\frac{2}{3}, \frac{2}{3})$$

$$(\frac{2}{3} + \epsilon, \frac{2}{3})$$

Manipulating Output (Possible)

Before i^* : $f(\hat{x}, y)$

			y_1	y_2
$1/3-\epsilon$	1/3	x_1	**0**	**1**
$1/3$	1/3	x_2	**1**	**0**
$1/3+\epsilon$	1/3	x_3	**1**	**1**

$$\left(\frac{2}{3}, \frac{2}{3}\right)$$

$$\left(\frac{2}{3} + \epsilon, \frac{2}{3}\right)$$

Manipulating Output (Impossible)

Before $i^*: f(\hat{x}, y)$

		y_1	y_2
1/2	x_1	**0**	**1**
1/2	x_2	**1**	**0**
		(1/2,	**1/2)**

Manipulating Output (Impossible Function)

Before i^* : $f(\hat{x}, y)$

		y_1	y_2
1/2	x_1	**0**	**1**
1/2	x_2	**1**	**0**
		(1/2,	**1/2)**
		(1/2+ϵ	**1/2)**

Manipulating Output (Impossible Function)

Before i^* : $f(\hat{x}, y)$

			y_1	y_2
1/2	**1/2**	x_1	**0**	**1**
1/2+ϵ	**1/2**	x_2	**1**	**0**

(1/2, 1/2)

(1/2+ϵ 1/2)

"The Power of the Ideal Adversary"

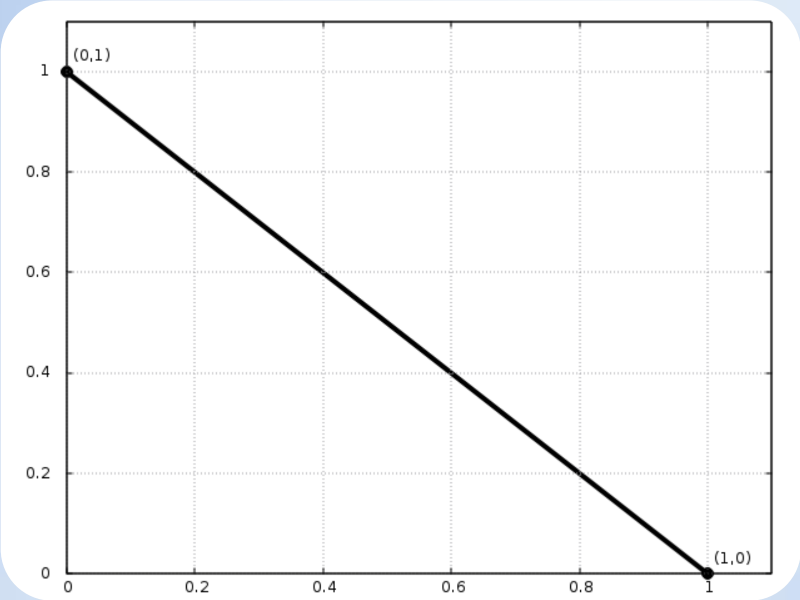
	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**

$(1-p, p)$

	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**
x_3	**1**	**1**

$(1-p_1, 1-p_2)$

“The Power of the Ideal Adversary”



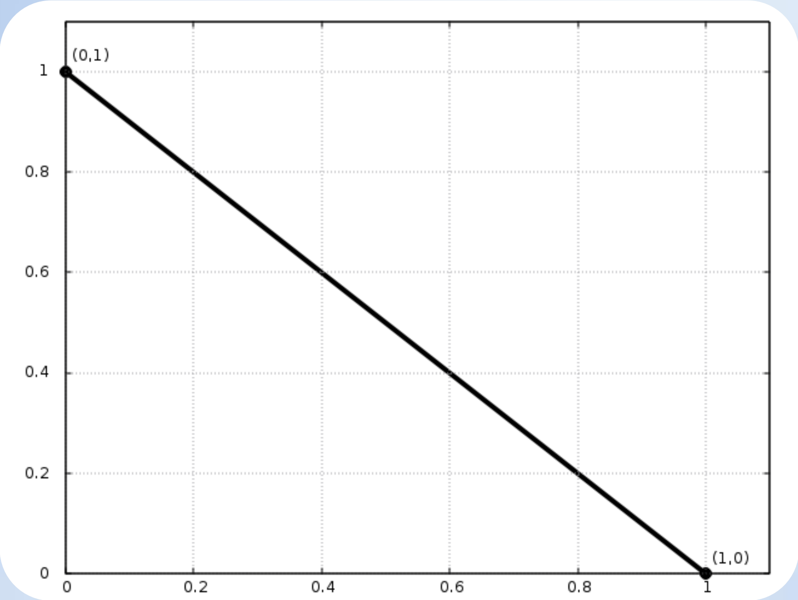
	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**

$$(1-p, p)$$

	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**
x_3	**1**	**1**

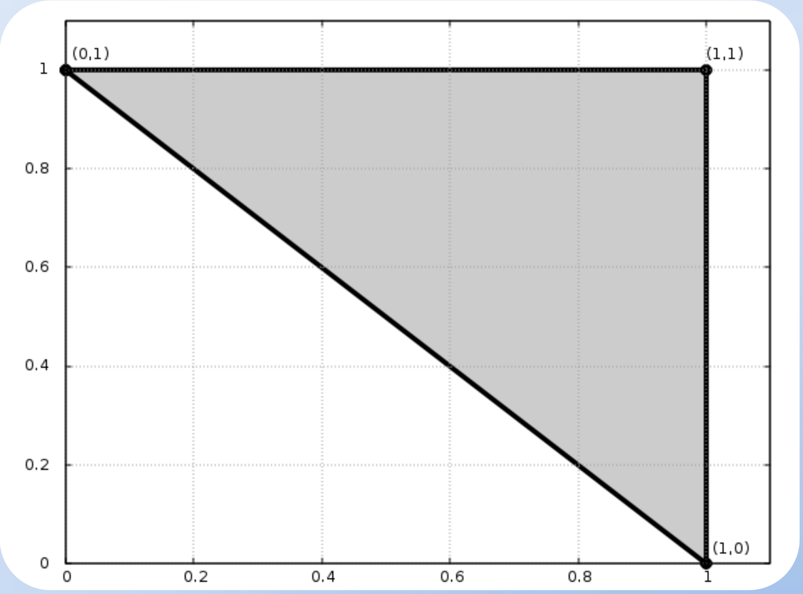
$$(1-p_1, 1-p_2)$$

"The Power of the Ideal Adversary"



	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**

$$(1-p, p)$$



	y_1	y_2
x_1	**0**	**1**
x_2	**1**	**0**
x_3	**1**	**1**

$$(1-p_1, 1-p_2)$$

Two Observations

1) **General for multiparty computation:**
   "The power of the ideal adversary"
   – Geometric representation
2) **Specific for the [GHKL08] protocol:**
   Adding more rounds – less to correct!

Second Observation: Back to the Protocol

REAL Before i^*:

$f(\hat{x}, y)$ for uniform $\hat{x}$ (1/3,1/3,1/3)

⇒(2/3, 2/3)

$E(R) = 5$

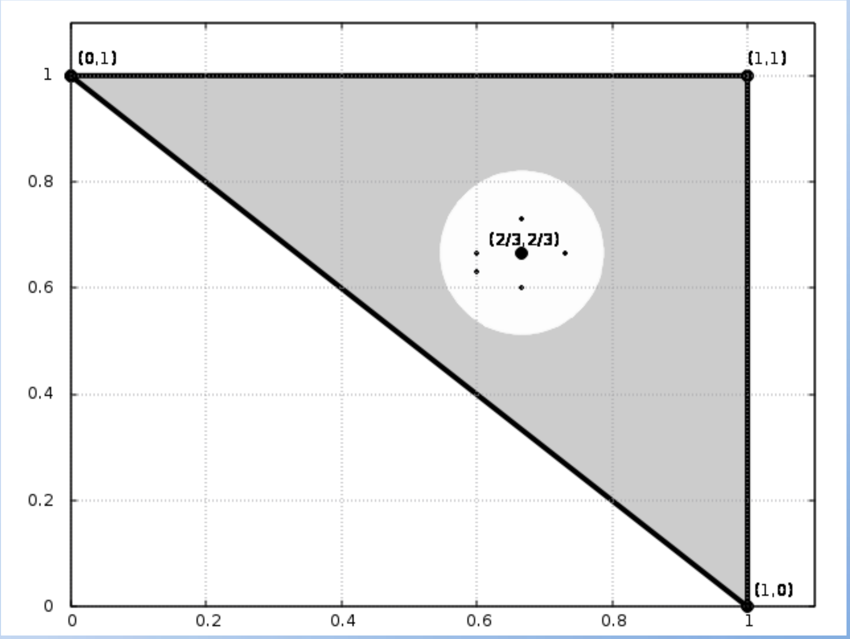
Input	a_i	$\widetilde{X}=(x_1,x_2,x_3)$	Output
x_1	0	(0, 1/3, 2/3)	(1, 2/3)
x_1	1	(1/3, 1/2, 1/6)	(2/3, 1/2)
x_2	0	(1/3, 0, 2/3)	(2/3, 1/2)
x_2	1	(1/2, 1/3, 1/6)	(1/2, 2/3)
x_3	0	(-,-,-)	(-,-)
x_3	1	(1/3, 1/3, 1/3)	(2/3, 2/3)

$E(R) = 100$

Input	a_i	$\widetilde{X}=(x_1,x_2,x_3)$	Output
x_1	0	(0.32, 0.33, 0.34)	(0.68, 0.67)
x_1	1	(0.36, 0.34, 0.32)	(0.67, 0.659)
x_2	0	(0.36, 0.31, 0.34)	(0.66, 0.68)
x_2	1	(0.34, 0.33, 0.32)	(0.65, 0.66)
x_3	0	(-,-,-)	(-,-)
x_3	1	(0.33, 0.33, 0.32)	(0.67, 0.67)

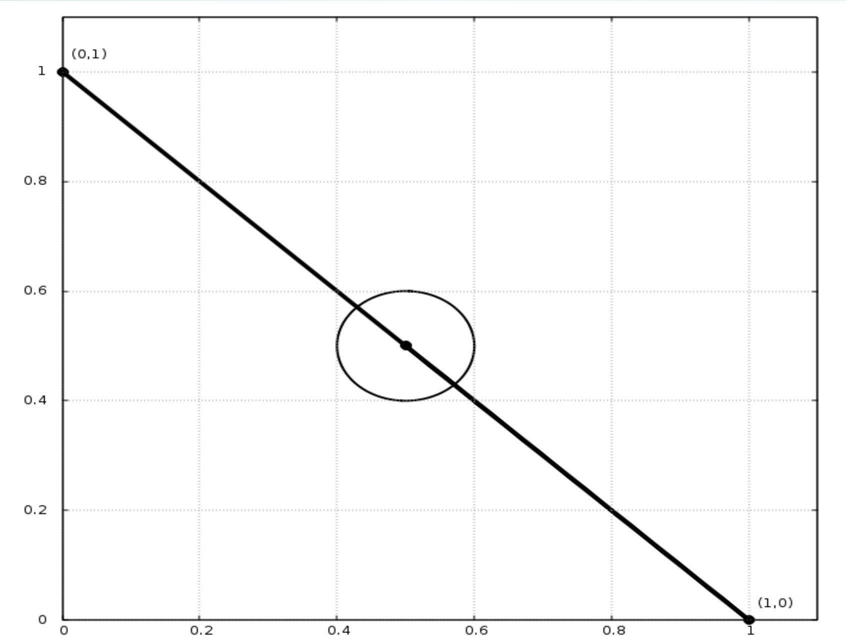
All points that the simulator needs are inside some "ball"

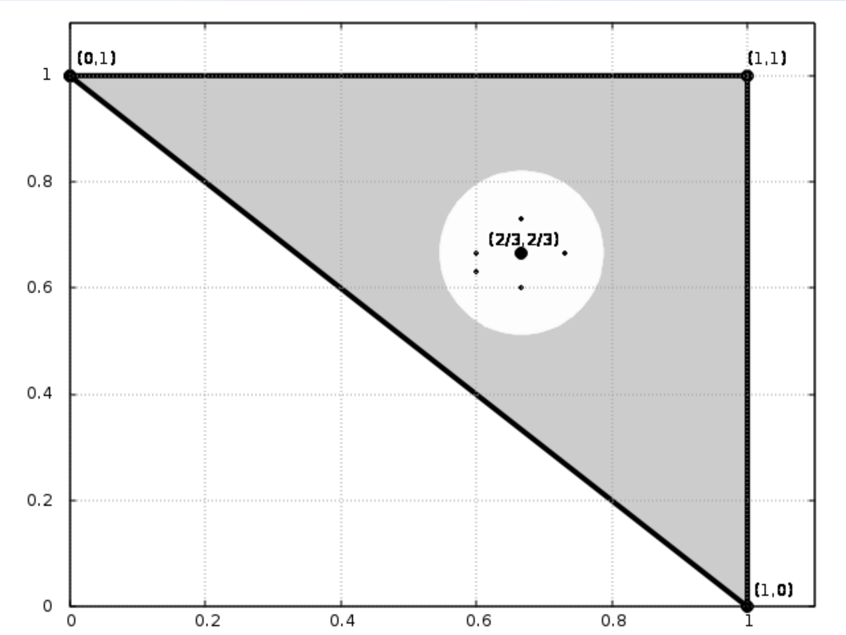
- **The center** – the output distribution of REAL
- **The radius** – a function of number of rounds



All points that the simulator needs are inside some “ball”

- **The center** – the output distribution of REAL
- **The radius** – a function of number of rounds





Full-Dimensional Functions

- Let $f: \{x_1, \dots, x_\ell\} \times \{y_1, \dots, y_m\} \to \{0,1\}$
- Consider the ℓ points $X_1, \dots, X_\ell$ in $\mathbb{R}^m$ (the "rows" of the matrix)

Full-Dimensional Functions

- Let $f: \{x_1, \ldots, x_\ell\} \times \{y_1, \ldots, y_m\} \rightarrow \{0,1\}$
- Consider the ℓ points $X_1, \ldots, X_\ell$ in $\mathbb{R}^m$ (the “rows” of the matrix)

Definition

If the geometric object defined by $X_1, \ldots, X_\ell \in \mathbb{R}^m$ is of dimension m,
Then the function is **full-dimensional**

Our Main Theorem

Theorem

If f is of **full-dimension**, then it can be computed with complete fairness

Our Main Theorem

Theorem

If f is of **full-dimension**, then it can be computed with complete fairness

Proof:

- We use the protocol of [GHKL08]

Our Main Theorem

Theorem

If f is of **full-dimension**, then it can be computed with complete fairness

Proof:

- We use the protocol of [GHKL08]
- We show that all the points that the simulator needs are inside a small “ball”

Our Main Theorem

Theorem

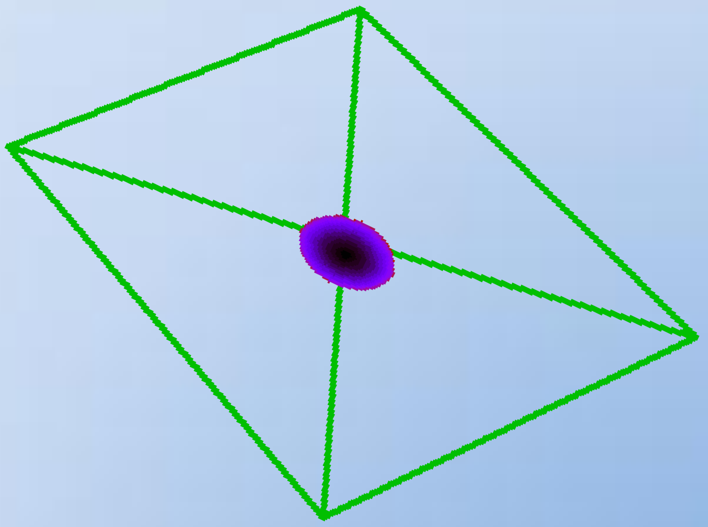
If f is of **full-dimension**, then it can be computed with complete fairness

Proof:

- We use the protocol of [GHKL08]
- We show that all the points that the simulator needs are inside a small "ball"
- The ball is embedded inside the geometric object defined by the function

Example in Higher Dimension

	y_1	y_2	y_3
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	1	1	1



Full Dimensional and Hyperplanes

- In $\mathbb{R}^2$ - all points do not lie on a single **LINE**
- In $\mathbb{R}^3$ - all points do not lie on a single **PLANE**
- …
- In $\mathbb{R}^m$ - all points do not lie on a single **HYPERPLANE**

Not Full-Dimensional

- In $\mathbb{R}^2$ - (z_1, z_2)

  $\exists(q_1, q_2, \delta) \in \mathbb{R}$ s.t. $q_1 z_1 + q_2 z_2 = \delta$?
- In $\mathbb{R}^3$ - (z_1, z_2, z_3)

  $\exists(q_1, q_2, q_3, \delta) \in \mathbb{R}$ s.t. $q_1 z_1 + q_2 z_2 + q_3 z_3 = \delta$?

Equivalent Representations

- Full-dimensional function
- The function is *right-unbalanced*:
  - For every non-zero $\boldsymbol{q} \in \mathbb{R}^m$, $\delta \in \mathbb{R}$ it holds that:

$$M_f \cdot \boldsymbol{q} \neq \delta \cdot \mathbf{1}$$

Equivalent Representations

- Full-dimensional function
- The function is *right-unbalanced*:
  - For every non-zero $\boldsymbol{q} \in \mathbb{R}^m$, $\delta \in \mathbb{R}$ it holds that:

$$M_f \cdot \boldsymbol{q} \neq \delta \cdot \mathbf{1}$$

Easy to Check Criterion:

No solution $\boldsymbol{q}$ for: $M_f \cdot \boldsymbol{q} = \mathbf{1}$

Only trivial solution for: $M_f \cdot \boldsymbol{q} = \mathbf{0}$

Balanced with respect to probability vector: IMPOSSIBLE!

Balanced with respect to probability vector: IMPOSSIBLE!

Unbalanced with respect to arbitrary vectors: FAIR!

Balanced with respect to probability vector: IMPOSSIBLE!

Unbalanced with respect to probability vector, balanced with respect to arbitrary vectors:

- **If the hyperplanes do not contain the origin:**
  cannot be computed using [GHKL08]
  (with particular simulation strategy)

- **If the hyperplanes contain the origin:**
  not characterized (sometimes the GHKL protocol is possible)

Unbalanced with respect to arbitrary vectors: FAIR!

CONCLUSIONS

On the Value P_d

P_d: The probability that a 0/1 matrix is singular?

On the Value P_d

- **P_d: The probability that a 0/1 matrix is singular?**
  - **Conjecture:** $(1/2+o(1))^d$
    (roughly the probability to have two rows that are the same)
  - **Komlos (67):**
    $$0.999^d$$
  - **Tao and Vu [STOC 05]:**
    $$(3/4+o(1))^d$$
  - **Best known today** [Vu and Hood 09]:
    $$(1/\sqrt{2}+o(1))^d$$

On the Value P_d

- **P_d: The probability that a 0/1 matrix is singular?**
  - **Conjecture:** $(1/2+o(1))^d$
    (roughly the probability to have two rows that are the same)
  - **Komlos (67):**
    $$0.999^d$$
  - **Tao and Vu [STOC 05]:**
    $$(3/4+o(1))^d$$
  - **Best known today** [Vu and Hood 09]:
    $$(1/\sqrt{2}+o(1))^d$$

On the Value P_d

- **P_d: The probability that a 0/1 matrix is singular?**
  - **Conjecture:** $(1/2+o(1))^d$
    (roughly the probability to have two rows that are the same)
  - **Komlos (67):**
    $$0.999^d$$
  - **Tao and Vu [STOC 05]:**
    $(3/4+o(1))^d$
  - **Best known today** [Vu and Hood 09]:
    $(1/\sqrt{2}+o(1))^d$

d	P_d
1	0.5
5	0.627
10	0.297
15	0.047
20	0.0025
25	0.0000689
30	0.0000015

What is the Probability that...

- The $d + 1$ random 0/1-points in $\mathbb{R}^d$ defines full-dimensional geometric object?
  - $1 - P_d$ (tends to 1)
- d points in $\mathbb{R}^d$ define hyperplane that passes through **0**,**1?**
  - $4P_d$ (tends to 0)

What is the Probability that...

- The $d+1$ random 0/1-points in $\mathbb{R}^d$ defines full-dimensional geometric object?
  - **1- P_d** **(tends to 1)**
- d points in $\mathbb{R}^d$ define hyperplane that passes through **0**,**1?**
  - **$4P_d$** **(tends to 0)**

- Almost all functions with $|X| \neq |Y|$:
  can be computed with **complete fairness**
- Almost all functions with $|X| = |Y|$:
  cannot be computed with [GHKL08] framework

What's Else in the Paper?

- **$d \times d$ functions with monochromatic input**
  - Define hyperplanes that pass through **0** or **1**
  - Almost always – possible
- **Asymmetric functions**
  - $f(x, y) = (f_1, f_2)$
  - If f_1 or f_2 are full-dimensional $\Rightarrow$ possible!
- **Non-binary outputs $f: X \times Y \to \Sigma$**
  - General criteria, holds when $|X|/|Y| > |\Sigma| - 1$

	y_1	y_2
x_1	0	1
x_2	1	0
x_3	1	1
x_4	2	0
x_5	1	2

What's Next?

- The characterization is not complete
- We have a better understanding of the "power" of the **ideal** world adversary
- We have no real understanding of the "power" of the **real**-world adversary
- Open problem:
  - Finalize the characterization!
  - Almost all functions with $|X| = |Y|$ are unknown

What's Next?

- The characterization is not complete
- We have a better understanding of the "power" of the **ideal** world adversary
- We have no real understanding of the "power" of the **real**-world adversary
- Open problem:
  - Finalize the characterization!
  - Almost all functions with $|X| = |Y|$ are unknown

Thank you!