Towards Characterizing Complete Fairness in Secure Two-Party Computation

Gilad Asharov

TCC 2014
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
If the adversary learns the output, then all parties should learn also

- In some sense, parties receive outputs simultaneously
If the adversary learns the output, then all parties should learn also

– In some sense, parties receive outputs simultaneously
• **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*?
Difficult 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

\[ f(x, y) \]
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
• Take a fair protocol
• Remove the last round
  -> still fair protocol
• Continue the process..
• We stay with an empty protocol
• Take a fair protocol
• Remove the last round
  -> still fair protocol
• Continue the process..
• We stay with an empty protocol
• Take a fair protocol
• Remove the last round
  -> still fair protocol
• Continue the process..
• We stay with an empty protocol
• Take a fair protocol
• Remove the last round
  -> still fair protocol
• Continue the process..
• We stay with an empty protocol
• In **1986**, Cleve showed that fairness is **impossible** in general (two party)
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
• 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

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Since 1986, the accepted belief was that *nothing* non-trivial can be computed fairly.
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]...
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.
Gordon, Hazay, Katz and Lindell [STOC08] showed that there exist some non-trivial functions that can be computed with complete fairness!
Gordon, Hazay, Katz and Lindell [STOC08] showed that there exist some non-trivial functions that can be computed with 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

![Matrix](image)

**Table:**

<table>
<thead>
<tr>
<th></th>
<th>$Y_1$</th>
<th>$Y_2$</th>
<th>$Y_3$</th>
<th>$Y_4$</th>
<th>$Y_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_5$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Matrix:**

<table>
<thead>
<tr>
<th></th>
<th>$Y_1$</th>
<th>$Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$X_2$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$X_3$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
A fundamental question:

What functions can and cannot be securely computed with complete fairness?
• A fundamental question:

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

• Impossibility: Cleve
A fundamental question:

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

- Impossibility: Cleve
- Only few examples of functions that are possible
• 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]
\[ f : X \times Y \to \{0, 1\} \]
with \(|X| \neq |Y|\)
$f : X \times Y \rightarrow \{0,1\}$
with $|X| \neq |Y|$
Set Membership

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

Set Membership

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

Private Evaluation of a Boolean Function

- **X input:** \( g \in F \) \( (F = \{g: \Omega \to \{0,1\}\}) \)
- **Y input:** \( y \in \Omega \)
- The function \( f(g, y) = g(y) \)
Private Matchmaking:
- X holds set of preferences ("what I am looking for")
- Y holds a profile ("who I am")
- Output: Does Y match X
Private Matchmaking:
- X holds set of preferences (“what I am looking for”)
- Y holds a profile (“who I am”)
- Output: Does Y match X

\[ A \subseteq B: \]
- X holds \( A \subseteq \Omega \)
- Y holds \( B \subseteq \Omega \)
- Output: \( A \subseteq B? \)
Private Matchmaking:
- X holds set of preferences (“what I am looking for”)
- Y holds a profile (“who I am”)
- Output: Does Y match X

\[ A \subseteq B: \]
- 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

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]
Impossible

\[ A = B \]
implies coin-tossing

[ALR13]
### Examples

<table>
<thead>
<tr>
<th>Impossible</th>
<th>Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A = B$</td>
<td>$A \subseteq B$</td>
</tr>
</tbody>
</table>

Implies coin-tossing [ALR13]
Impossible

\[ A = B \]

implies coin-tossing

[ALR13]

Unknown

not coin-tossing

not [GHKL08]*

Possible

\[ A \subseteq B \]
A Full Characterization of Functions that Imply Fair Coin Tossing and Ramifications to Fairness

Asharov, Lindell, Rabin

TCC 2013
The coin-tossing functionality is impossible:

\[ f(\lambda, \lambda) = (U, U) \]

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

- both parties agree on the same uniform bit
- no party can bias the result
The coin-tossing functionality is impossible:

\[ f(\lambda, \lambda) = (U, U) \]

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

- 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

Question:
Assume a fair protocol for the XOR function. How can we use it to toss a coin?
Assume a fair protocol for the XOR function. How can we use it to toss a coin?

Each party chooses a uniform bit, then XOR them.
Why Does it Work?

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

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

\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}
Why Does it Work?

$$\Pr[\text{output } = 1] = (p_1 \quad p_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

distribution over the inputs of $X$

distribution over the inputs of $Y$

$$\begin{pmatrix} 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \frac{1}{2}$$
Why Does it Work?

\[ \Pr[\text{output} = 1] = (p_1 \quad p_2) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ q_1 \\ q_2 \end{pmatrix} \]

distribution over the inputs of X

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

distribution over the inputs of Y

\[
\begin{pmatrix}
0 & 1 \\
1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
\frac{1}{2} \\
1/2 \\
\end{pmatrix}
\]
Why Does it Work?

\[
\text{Pr[output } = 1]\] = \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 \(X\)

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

distribution over the inputs of \(Y\)

\[
\begin{pmatrix} p_1 & p_2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \begin{pmatrix} p_1 & p_2 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} = \frac{1}{2}
\]
if there exist probability vectors \( p = (p_1, \ldots, p_m) \), \( q = (q_1, \ldots, q_\ell) \) and \( 0 < \delta < 1 \) s.t:

\[
p \cdot M_f = \delta \cdot 1_\ell \quad \text{AND} \quad M_f \cdot q^T = \delta \cdot 1_m^T
\]

\( f \) is \( \delta \) balanced
The Property

If \( f \) is \( \delta \)-balanced then it implies fair coin-tossing.

**Theorem**

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

\[
\begin{align*}
    p \cdot M_f &= \delta \cdot 1_\ell \\
    M_f \cdot q^T &= \delta \cdot 1_m^T
\end{align*}
\]
Other Examples

Balanced Functions:

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\quad
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\quad
\begin{pmatrix}
1 & 0 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

Unbalanced Functions:

\[
\begin{pmatrix}
1 & 0 \\
0 & 1 \\
1 & 1
\end{pmatrix}
\]

(left-balanced, right-unbalanced)
Other Examples

Balanced Functions:

\[
\begin{pmatrix}
1 & 0 \\
0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

Unbalanced Functions:

\[
\begin{pmatrix}
1 & 0 \\
0 & 1 \\
1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 1 \\
1 & 1 \\
\end{pmatrix} (1 - p) = \begin{pmatrix}
p \\
1 - p \\
\end{pmatrix}
\]

(left-balanced, right-unbalanced)
if $f$ is not $\delta$-balanced for any $0 < \delta < 1$, then it does not imply coin tossing*
if $f$ is not $\delta$-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.
Theorem

if \( f \) is not \( \delta \)-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 $\delta$-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**.
Theorem

if \( f \) is not \( \delta \)-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
Gordon, Hazay, Katz and Lindell [STOC08] presented a general protocol and proved that a particular function can be computed using this protocol.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th></th>
<th>$Y_1$</th>
<th>$Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$X_2$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$X_3$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Question:

What functions can be computed using this protocol?
• Almost all functions with $|X| \neq |Y|$: can be computed using the protocol

• Almost all functions with $|X| = |Y|$: 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
• 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 $P_y$
• 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

Trusted Party
The Definition

Trusted Party
The Definition

Trusted Party

Px \rightarrow x \rightarrow \text{Trusted Party} \rightarrow y \rightarrow Py
The Definition

Trusted Party

\[ f(x, y) \]
The Definition

Trusted Party

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

<table>
<thead>
<tr>
<th></th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/3 x_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$1/3 x_2$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$1/3 x_3$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$(\frac{2}{3}, \frac{2}{3})$
Before $i^* : f(\hat{x}, y)$

<table>
<thead>
<tr>
<th>$1/3$</th>
<th>$\chi_1$</th>
<th>$\chi_2$</th>
<th>$\chi_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$0$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td></td>
<td>$1$</td>
<td>$0$</td>
<td>$1$</td>
</tr>
<tr>
<td></td>
<td>$1$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
</tbody>
</table>

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

$\left( \frac{2}{3} + \epsilon, \frac{2}{3} \right)$
Before $i^* : f(\hat{x}, y)$

| $1/3 - \epsilon$ | $1/3$ | $x_1$ | $0$ | $1$ |
| $1/3$ | $x_2$ | $1$ | $0$ |
| $1/3 + \epsilon$ | $x_3$ | $1$ | $1$ |
| $(2/3, 2/3)$ |
| $(2/3 + \epsilon, 2/3)$ |
Manipulating Output (Possible)

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

<table>
<thead>
<tr>
<th>$1/3 - \epsilon$</th>
<th>$1/3$</th>
<th>$1/3$</th>
<th>$1/3 + \epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>$Y_1$</th>
<th>$Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>$x_1$</td>
<td>0</td>
</tr>
<tr>
<td>1/2</td>
<td>$x_2$</td>
<td>1</td>
</tr>
</tbody>
</table>

$(1/2, \ 1/2)$
Manipulating Output (Impossible Function)

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

<table>
<thead>
<tr>
<th></th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/2$</td>
<td>$x_1$</td>
<td>0</td>
</tr>
<tr>
<td>$1/2$</td>
<td>$x_2$</td>
<td>1</td>
</tr>
</tbody>
</table>

$(1/2, 1/2)$

$(1/2 + \epsilon, 1/2)$
Manipulating Output (Impossible Function)

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

<table>
<thead>
<tr>
<th></th>
<th>$Y_1$</th>
<th>$Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$\frac{1}{2}$, $\frac{1}{2}$

$\frac{1}{2} + \epsilon$, $\frac{1}{2}$

$(\frac{1}{2}, \frac{1}{2})$

$(\frac{1}{2} + \epsilon, \frac{1}{2})$
"The Power of the Ideal Adversary"

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

$\begin{pmatrix} (1 - p, p) \end{pmatrix}$

<table>
<thead>
<tr>
<th>$x_2$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

$\begin{pmatrix} (1 - p_1, 1 - p_2) \end{pmatrix}$

<table>
<thead>
<tr>
<th>$x_3$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

“The Power of the Ideal Adversary”

\[
\begin{align*}
\begin{array}{c|cc}
 x_1 & Y_1 & Y_2 \\
 0 & 1 & 0 \\
 (1 - p, p) \\
 x_2 & 1 & 0 \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{c|cc}
 x_1 & Y_1 & Y_2 \\
 0 & 1 & 0 \\
 x_2 & 1 & 1 \\
 (1 - p_1, 1 - p_2) \\
 x_3 & 0 & 0 \\
\end{array}
\end{align*}
\]
“The Power of the Ideal Adversary”

\[ (1 - p, p) \]

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

\[(1 - p_1, 1 - p_2)\]
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!
REAL Before $i^*$:
$f(\hat{x}, y)$ for uniform $\hat{x}$ (1/3, 1/3, 1/3) ⇒(2/3, 2/3)

$E(R) = 5$  
$E(R) = 100$

<table>
<thead>
<tr>
<th>Input</th>
<th>$a_i$</th>
<th>$\vec{x}=(x_1, x_2, x_3)$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>(0, 1/3, 2/3)</td>
<td>(1, 2/3)</td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>(1/3, 1/2, 1/6)</td>
<td>(2/3, 1/2)</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>(1/3, 0, 2/3)</td>
<td>(2/3, 1/2)</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>(1/2, 1/3, 1/6)</td>
<td>(1/2, 2/3)</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>(-,-,-)</td>
<td>(-,-)</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>(1/3, 1/3, 1/3)</td>
<td>(2/3, 2/3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>$a_i$</th>
<th>$\vec{x}=(x_1, x_2, x_3)$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>(0.32, 0.33, 0.34)</td>
<td>(0.68, 0.67)</td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>(0.36, 0.34, 0.32)</td>
<td>(0.67, 0.659)</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>(0.36, 0.31, 0.34)</td>
<td>(0.66, 0.68)</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>(0.34, 0.33, 0.32)</td>
<td>(0.65, 0.66)</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>(-,-,-)</td>
<td>(-,-)</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>(0.33, 0.33, 0.32)</td>
<td>(0.67, 0.67)</td>
</tr>
</tbody>
</table>
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
• Let $f: \{x_1, ..., x_\ell\} \times \{y_1, ..., y_m\} \rightarrow \{0,1\}$
• Consider the $\ell$ points $X_1, ..., X_\ell$ in $\mathbb{R}^m$ (the “rows” of the matrix)
Let $f: \{x_1, \ldots, x_\ell\} \times \{y_1, \ldots, y_m\} \rightarrow \{0,1\}$

Consider the $\ell$ 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**
If $f$ is of **full-dimension**, then it can be computed with complete fairness
If $f$ is of **full-dimension**, then it can be computed with complete fairness.

Proof:

- We use the protocol of [GHKL08]
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”
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

<table>
<thead>
<tr>
<th></th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
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. $q_1z_1 + q_2z_2 = \delta$?
- In $\mathbb{R}^3$ - $(z_1, z_2, z_3)$
  $\exists (q_1, q_2, q_3, \delta) \in \mathbb{R}$ s.t. $q_1z_1 + q_2z_2 + q_3z_3 = \delta$?
• Full-dimensional function

• The function is right-unbalanced:
  – For every non-zero \( q \in \mathbb{R}^m, \delta \in \mathbb{R} \) it holds that:
  \[
  M_f \cdot q \neq \delta \cdot 1
  \]
• Full-dimensional function
• The function is right-unbalanced:
  – For every non-zero $q \in \mathbb{R}^m$, $\delta \in \mathbb{R}$ it holds that:
    $$M_f \cdot q \neq \delta \cdot 1$$

**Easy to Check Criterion:**

No solution $q$ for: $M_f \cdot q = 1$
Only trivial solution for: $M_f \cdot q = 0$
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!
$P_d$: 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]:**
    
    $(1/\sqrt{2}+o(1))^d$
• $P_d$: 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]: $(1/\sqrt{2}+o(1))^d$
• $P_d$: 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]:**
  $(1/\sqrt{2} + o(1))^d$

<table>
<thead>
<tr>
<th>d</th>
<th>$P_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.627</td>
</tr>
<tr>
<td>10</td>
<td>0.297</td>
</tr>
<tr>
<td>15</td>
<td>0.047</td>
</tr>
<tr>
<td>20</td>
<td>0.0025</td>
</tr>
<tr>
<td>25</td>
<td>0.0000689</td>
</tr>
<tr>
<td>30</td>
<td>0.0000015</td>
</tr>
</tbody>
</table>
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
• $d \times d$ 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 $f: X \times Y \rightarrow \Sigma$
  – General criteria, holds when $|X|/|Y| > |\Sigma| - 1$
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
• 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!