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

Gilad Asharov  
Bar-Ilan University

Yehuda Lindell  
Bar-Ilan University

Tal Rabin  
IBM Research

TCC 2013
Secure Multiparty Computation

• A set of parties with private inputs wish to compute some joint function of their inputs
• Parties wish to preserve some security properties. E.g., privacy and correctness
• Security must be preserved in the face of adversarial behavior by some of the participants, or by an external party
• The adversary receives an output if and only if the honest party receives an output
  – In some sense, parties receive outputs simultaneously
Coin-Tossing

- The coin-tossing functionality:
  \[ f(\lambda, \lambda) = (U, U) \]
  
  \((U\) is the uniform distribution over \(\{0,1\}\))
  
  - both parties agree on the same uniform bit
  - no party can bias the result

- In 1986, Cleve showed that it is impossible to construct a fair coin-tossing protocol
  
  - Intuitively, no simultaneous exchange, so one party always has more information about the result, and can abort and bias the result
• Gordon, Hazay, Katz and Lindell [STOC08] showed that there exist **some non-trivial** functions that can be computed with **complete fairness**!

  – Any protocol with no embedded XOR (essentially the less-than functionality)
  – Some specific functionalities with embedded XOR
A fundamental question:

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

The only known impossibility result for fairness today is still that of Cleve
What Do We Know About the World?

- coin-tossing

- trivial \(\checkmark\)
- less-than \(\checkmark\)
- functions that imply coin-tossing \(\times\)
- anything else?
- some functionalities with embedded XOR [GHKL] \(\checkmark\)
Characterizing Fairness

• Which Boolean functions with finite domain can be computed with complete fairness?

• Can we characterize the functions via a property such that:
  – If the function satisfies the property: it can be computed fairly
  – If the function does not satisfy the property: it cannot be computed fairly
• We give a simple property (a criterion) such that
• If the function satisfies the property – it implies coin-tossing
  – Thus, it cannot be computed with complete fairness
• If the function does not satisfy the property – it does not imply coin-tossing*
  – We know exactly what Cleve’s impossibility rules out
  – Proving impossibility for other functions requires a new proof (cannot be reduced to Cleve)
What Do We Know About the World?

Deterministic Boolean Functions with Finite Domain

- trivial ✓
- less-than ✓
- anything else?
- Functions that imply coin-tossing ✗
- some functionalities with embedded XOR [GHKL] ✓
What About This Function?

<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>
### What About This Function?

<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>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What About This Function?

<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>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What About This Function?

<table>
<thead>
<tr>
<th>1/2</th>
<th>x_1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>x_2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>x_3</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What About This Function?

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1/2</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1/2 1/2 1/2
The Protocol – Malicious Y

pick $x \in \{x_1, x_2, x_3\}$ according to distr. $(1/2, 1/2, 0)$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$f(x,y) = \frac{1}{2}$
<table>
<thead>
<tr>
<th></th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$\gamma_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
### What About Party Y?

<table>
<thead>
<tr>
<th></th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/2$</td>
<td>$1/2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$1/2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$1/2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$1/2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
The Overall Protocol

pick \( x \in \{x_1, x_2, x_3\} \) according to distr. \((1/2, 1/2, 0)\)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

pick \( y \in \{y_1, y_2, y_3\} \) according to distr. \((1/2, 0, 1/2)\)

\[
\text{Pr}[f(x, y) = 1] = \frac{1}{2}
\]
Another Point of View

<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>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Another Point of View

\[
\begin{pmatrix}
\ p_1 \\
\ p_2 \\
\ p_3
\end{pmatrix}
\begin{pmatrix}
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
q_1 \\
q_2 \\
q_3
\end{pmatrix}
\]

\[= \Pr[\text{output} = 1] \]
Another Point of View

\[
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & 0 \\
1 & 0 & 1 \\
1 & 0 & 0 \\
\end{pmatrix}
\begin{pmatrix}
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
q_1 \\
q_2 \\
q_3 \\
\end{pmatrix} =
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\end{pmatrix}
\begin{pmatrix}
q_1 \\
q_2 \\
q_3 \\
\end{pmatrix} = \frac{1}{2}
\]

\[
\begin{pmatrix}
p_1 & p_2 & p_3 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 1 & 0 \\
\end{pmatrix}
\begin{pmatrix}
1/2 \\
0 \\
1/2 \\
\end{pmatrix} =
\begin{pmatrix}
p_1 & p_2 & p_3 \\
1/2 & 1/2 & 1/2 \\
\end{pmatrix} = \frac{1}{2}
\]
Generalizing the Above: Definitions

\( f \) is \( \delta_1 \)-left balanced

if there exists a probability vector \( p = (p_1, \ldots, p_m) \), \( 0 \leq \delta_1 \leq 1 \)
such that: \( p \cdot M_f = \delta_1 \cdot 1_{\ell} \)

\( f \) is \( \delta_2 \)-right balanced

if there exists a probability vector \( q = (q_1, \ldots, q_{\ell}) \), \( 0 \leq \delta_2 \leq 1 \)
such that: \( M_f \cdot q^T = \delta_2 \cdot 1_{m}^T \)

\( f \) is \( \delta \)-balanced

if \( f \) is \( \delta \)-left balanced and \( \delta \)-right balanced
Our Main Theorem

- If $f$ is $\delta$-balanced for some $0 < \delta < 1$, then it implies coin-tossing

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

If \( f \) is \( \delta \)-balanced for some \( 0 < \delta < 1 \), then it implies coin-tossing.

Proof:

\( f \) is \( \delta \)-balanced \( \Rightarrow \) coin tossing for \( \delta \)-coin

Apply von-Neumann’s method to toss a fair-coin.
The Impossibility Result

**Theorem**

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

- We show that there does not exist a fair coin-tossing protocol in the $f$-hybrid model
  - For any coin-tossing protocol in the $f$-hybrid model, there exists an (inefficient) adversary that can bias the result
- Unlike Cleve – the parties have some **simultaneous exchange**. Thus, a completely different argument is needed
Impossibility in the OT-hybrid model

- The adversary is *inefficient*
  - It computes the distributions over all possible random coins of an honest $X$
  - This computation can be approximated given an $NP$-oracle
- We do not know how to construct an *efficient* adversary
- Impossibility still holds if the parties have an ideal OT
  - Embedded OR implies OT [Kilian 91]
  - A function that doesn’t contain an embedded OR is $1/2$-balanced
A malicious $Y$ can always bias the probability to get 1 in a single invocation!
We can assume that the protocol consists only of invocations of $f$. 
The Protocol Transcript Tree

\[(r_1, r_2)\]

\[(r_1, r_2): x_i, y_j: 1\]

\[(r_1, r_2): x_k, y_k: 0\]
The Protocol Transcript Tree

distributions over the inputs

$(\alpha_1, \ldots, \alpha_m)$
$(\beta_1, \ldots, \beta_\varepsilon)$

probability of $\text{coin}=1$ conditioned on reaching $\nu$

probability that the output is 1 in this invocation

$p_\nu$

$0$

$1 - \gamma^\nu$

$\gamma^\nu$

$(r_1, r_2): x_i, y_j: 1$

$1$

$0$

$1$

$0$

$1$
The adversary acts honestly but searches for a “jump” between probabilities in parent and children.

We show that in any execution, such a “jump” exists.
• We also study the case of a **fail-stop** adversary
  – Follows the protocol specifications but may abort prematurely
• Unclear how to model fail-stop in the ideal world
  – Is the simulator allowed to change the corrupted party’s input?
• We consider two possible definitions and study fairness in *both* cases
• We give a simple property (a criterion) s.t.:
  – If the function satisfies the property, it implies coin-tossing
  – If the function does not satisfy the property, it does not imply coin-tossing
• We consider the same question for fail-stop adversary
• This is an important step forward towards understanding fair secure computation

Thank You!!
Every path (execution) from root to leaf has such a “jump”:

- The probability in the root is $1/2$
- The probability in each leaf is either 0 or 1

End of Proof Sketch