Fair Secure Computation
(or how can I gain strategic advantage by breaking fairness)

Alptekin Küpçü
Koç University
Fair Secure Computation

ALPTEKİN KÜPÇÜ
Assistant Professor of Computer Science and Engineering
Secure Multi-Party Computation

\[ X_1 \quad X_2 \]
\[ X_3 \quad X_4 \]
\[ X_5 \quad X_6 \]
Secure Multi-Party Computation
Secure Multi-Party Computation

12 May 2016

Alptekin Küpçü - Koç University
Ideal World

X₁

X₂

X₃

X₄

X₅

X₆
Ideal World
Real World
Simulator

\[ X_1 \rightarrow X \rightarrow X_2 \]

\[ X_3 \rightarrow X \rightarrow X_4 \]

\[ X_5 \rightarrow X \rightarrow X_6 \]
Fairness Impossible in General

- Assume a trusted Arbiter is available
  - Only trusted for fairness, not security
    - May collude with players
    - Should not learn input/output
  - Optimistically employed
  - Must be efficient (otherwise bottleneck)
Fairness Impossible in General

- Asssume a trusted Arbiter is available
  - Only trusted for fairness, not security
    - May collude with players
    - Should not learn input/output
  - Optimistically employed
  - Must be efficient (otherwise bottleneck)

- Ideal TTP
- Real Arbiter
Fair and Secure Computation

- Fairness extensions and Arbiter resolutions must be simulated
Simulating Fairness

SECURE 2PC SIMULATION

FAIRNESS ARGUMENT
Simulating Fairness

SECURE and FAIR 2PC SIMULATION
Fair and Secure Computation

- Fairness extensions and Arbiter resolutions must be simulated
  - Otherwise the protocol may be insecure!
Fair and Secure Computation

- Fairness extensions and Arbiter resolutions must be simulated
  - Otherwise the protocol may be insecure!
- Simulator may contact only when fairness is guaranteed

12 May 2016
Simulator

\[ X_1 \rightarrow X_3 \rightarrow X_5 \rightarrow \text{Barbie} \rightarrow X_2 \rightarrow X_4 \rightarrow X_6 \]
Fair and Secure Computation

- Fairness extensions and Arbiter resolutions must be simulated
  - Otherwise the protocol may be insecure!
- Simulator may contact only when fairness is guaranteed
  - Otherwise real and ideal world outputs are distinguishable
Fair and Secure Computation

- Fairness extensions and Arbiter resolutions must be simulated
  - Otherwise the protocol may be insecure!
- Simulator may contact only when fairness is guaranteed
  - Otherwise real and ideal world outputs are distinguishable
- Arbiter cannot harm security
Our Solutions

<table>
<thead>
<tr>
<th># Participants</th>
<th># Rounds</th>
<th># Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>$n$</td>
<td>$O(1)$</td>
<td>$O(n^2)$</td>
</tr>
</tbody>
</table>

- OPTIMAL asymptotic performance
- Cut-and-choose or zero-knowledge
- Malicious or covert
- 2PC or MPC
Comparison

- Compared to related works, we provide
  - Optimal asymptotic performance
    - Constant round (not gradual release)
    - No broadcast
    - Arbiter load independent of the circuit size
  - Do not require an external payment mechanism
    - In a competitive corporate setting, how can one value some output that is unknown beforehand?
  - Full simulation proofs
  - Arbiter cannot harm security
    - Also proven via simulation
    - Only fairness is lost if Arbiter colludes with malicious parties
Our Papers

- **Reading**
  - Küpçü and Mohassel, FC 2016, *Fast Optimistically Fair Cut-and-Choose 2PC*

- **Funding Acknowledgements**
  - TÜBİTAK, the Scientific and Technological Research Council of TURKEY
  - COST Action IC1306 Cryptoaction
ALPTEKİN KÜPÇÜ
Assistant Professor of Computer Science and Engineering

http://crypto.ku.edu.tr