Advanced Tools from Modern Cryptography

Lecture 15
MPC: Complexity of Functions
Feasibility of General MPC

Given honest majority, or given OT as a setup:

- General MPC is possible with the highest security guarantee (information-theoretic, UC security)
- Variations: $t < n/3$ vs. $t < n/2 + \text{broadcast}$. Perfect vs. Statistical. Guaranteed output delivery vs. unfair.

<table>
<thead>
<tr>
<th>Otherwise:</th>
<th>Passive</th>
<th>Stand-alone</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPT</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Info. Th.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Impossibility of general MPC reveals “Cryptographic Complexity” of functions: some are more “complex” than the others.

In each security model, functionalities that admit MPC protocols without a setup form the least complex — a.k.a. trivial — functionalities in that model.
Trivial Functionalities: PPT Setting

General MPC under the assumption that there is a passive-secure protocol for OT (a.k.a. \textit{sh-OT}).

GMW: using ZK proofs (\textit{sh-OT} \Rightarrow \textit{OWF} \Rightarrow \textit{ZK}).

For \(n=2\), we have an explicit characterisation of trivial functions (splittable functions). Extends to \(n=3\) as well.

Open for \(n > 3\).

Recall: without honest majority, AND is impossible.

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Stand-alone</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPT</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Info. Th.</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
For n-party information-theoretic passive security, for each corruption threshold t: the **Privacy Hierarchy**

- All n-party functions appear till level $\lfloor (n-1)/2 \rfloor$ in this hierarchy (e.g., by Passive-BGW). Some reach level $n$: e.g., XOR or more generally, group addition. Level $n-1$ is same as level $n$.

- At all intermediate levels $t$, examples known to exist which are not in level $t+1$

**Open problem:** For all n, t, characterise the functions level $t$ of the n-party privacy hierarchy (or do it just for $t=n$)

- For $n=2$ we do have a characterisation
Trivial 2-Party Functionalities: Information-Theoretic

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Stand-alone</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPT</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Info. Th.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

For deterministic SFE:
Trivial $\iff$ Decomposable
Decomposable Function
(For simplicity will restrict to symmetric SFE)

Examples of Decomposable Functions

“Max” (no ties)

XOR

[(x+5y)/2]

Examples of Undecomposable Functions

AND

“Spiral”
Decomposable Function
Trivial 2-Party Functionalities: Information-Theoretic

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Stand-alone</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PPT</strong></td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Info. Th.</strong></td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

For deterministic SFE:
Trivial $\iff$ Decomposable

For deterministic SFE:
Trivial $\iff$ Uniquely Decomposable & Saturated

Open for randomized
**Decomposable Function**

Examples of Decomposable Functions

- Not Uniquely Decomposable
- Not Saturated

This strategy doesn't correspond to an input.
### Trivial 2-Party Functionalities: Information-Theoretic

<table>
<thead>
<tr>
<th></th>
<th>Passive</th>
<th>Stand-alone</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPT</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Info. Th.</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

For deterministic SFE:
- Trivial $\iff$ Decomposable
- Trivial $\iff$ Uniquely Decomposable & Saturated

Open for randomized
- Trivial $\iff$ Splittable

Trivial $\iff$ Splittable
We saw OT can be used to (passive- or UC-) securely realise any functionality.

i.e., any other functionality can be reduced to OT

The Cryptographic Complexity question:

Can F be reduced to G (for different reductions)?

F reduces to G: will write \( F \subseteq G \)

G \textbf{complete} if everything reduces to G

F \textbf{trivial} if F reduces to everything (in particular, to “null”)

Completeness
PPT Setting: Completeness

- PPT Passive security and PPT Standalone security
  - Under sh-OT assumption, all functions are trivial — and hence all are complete too!

- PPT UC security, n=2:
  - Recall, only a few (splittable) functionalities are trivial
  - Under sh-OT, turns out that in fact, every non-trivial functionality is complete
IT Setting: Completeness

- Information-Theoretic Passive security
- (Randomized) SFE: Complete ⇔ Not Simple
- What is Simple?
### Simple vs. Non-Simple

Edge \(((x,a),(y,b))\) exists iff \(f(x,y)=(a,b)\)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Simple: Each connected component is a biclique
IT Setting: Completeness

- Information-Theoretic Passive security
- (Randomized) SFE: Complete $\iff$ Not Simple
- What is Simple?
  - In the characteristic bipartite graph, each connected component is a biclique
  - If randomized, within each connected component $w(u,v) = w_A(u) \times w_B(v)$, where $u=(x,a)$, $v=(y,b)$ and $w(u,v) = \Pr[\text{out}=(a,b) \mid \text{in}=(x,y)]$
Simple vs. Non-Simple (Randomized)

Simple: within connected component
\[ w(u,v) = w_A(u) \cdot w_B(v) \]

Edge \((x,a),(y,b)\) weighted with
\[ \Pr[(a,b) \mid (x,y)] \]
where \(x, y\) inputs and \(a, b\) outputs

Rabin-OT
IT Setting: Completeness

- Information-Theoretic Passive security
  - (Randomized) SFE: Complete ⇔ Not Simple
- Information-Theoretic Standalone & UC security
  - (Randomized) SFE: Complete ⇔ Core is not Simple

What is the core of an SFE?

- SFE obtained by removing “redundancies” in the input and output space
  - E.g., AND with one-sided output is not simple, but its core is
A Map of 2-Party Functions

- **OR**
- **XOR**
- **max (no ties)**
- **(x+5y)/2**
- **Spiral**

**Saturated**

**Decomposable**

**Non-Simple**

**Splitable**