Round-Optimal and Communication-Efficient Multiparty Computation

Michele Ciampi, Rafail Ostrovsky, Hendrik Waldner, Vassilis Zikas
Multiparty Computation (MPC)
Multiparty Computation (MPC)
Multiparty Computation (MPC)

\[ f(x_1, x_2) \]

\[ f(x_1, x_2) \]
Multiparty Computation (MPC)

\[ f(x_1, x_2) \]
Multiparty Computation (MPC)

1. Number of Messages
Multiparty Computation (MPC)

1. Number of Messages
2. Size of Messages
Multiparty Computation (MPC)

Four rounds are necessary [GMPP16] and sufficient [CCG+19,BGJ+18, HHPV18]

1. Number of Messages
2. Size of Messages
Multiparty Computation (MPC)

Four rounds are necessary \[\text{[GMPP16]}\] and sufficient \[\text{[CCG+19, BGJ+18, HHPV18]}\]

1. Number of Messages ✓
2. Size of Messages
Multiparty Computation (MPC)

Four rounds are necessary [GMPP16] and sufficient [CCG+19, BGJ+18, HHPV18]

1. Number of Messages ✓
2. Size of Messages ←
Prior Work
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[BL18] [GS18]</td>
<td>OT</td>
<td></td>
<td>/</td>
</tr>
</tbody>
</table>
## Prior Work

<table>
<thead>
<tr>
<th>Round-Optimal Protocols</th>
<th>Semi-Honest</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BL18] [GS18]</td>
<td>OT</td>
<td>/</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Work</th>
<th>Assumptions</th>
<th>Comm. Compl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BL18]</td>
<td>OT</td>
<td>/</td>
</tr>
<tr>
<td>[GS18]</td>
<td>[BGJ+18]</td>
<td>DDH/Q-N Res.</td>
</tr>
</tbody>
</table>

Note: / denotes information not available or not applicable.
### Prior Work

<table>
<thead>
<tr>
<th>Round-Optimal Protocols</th>
<th>Semi-Honest</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>Assumptions</td>
<td>Comm. Compl.</td>
</tr>
<tr>
<td>[GS18]</td>
<td></td>
<td>[CCG+20] OT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Prior Work

<table>
<thead>
<tr>
<th>Semi-Honest</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Work</strong></td>
<td><strong>Work</strong></td>
</tr>
<tr>
<td>[BL18]</td>
<td>[BGJ+18]</td>
</tr>
<tr>
<td>[GS18]</td>
<td>DDH/Q-N Res.</td>
</tr>
<tr>
<td><strong>Assumptions</strong></td>
<td><strong>Assumptions</strong></td>
</tr>
<tr>
<td>OT</td>
<td>[CCG+20]</td>
</tr>
<tr>
<td>depth((f))</td>
<td>OT</td>
</tr>
<tr>
<td><strong>Comm. Compl.</strong></td>
<td><strong>Comm. Compl.</strong></td>
</tr>
<tr>
<td>(||)</td>
<td>(||)</td>
</tr>
</tbody>
</table>

**Round-Optimal Protocols**

**With Improved Comm. Compl.**
### Prior Work

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[GS18]</td>
<td></td>
<td></td>
<td>[CCG+20]</td>
<td>OT</td>
<td></td>
</tr>
<tr>
<td>With Improved Comm. Compl.</td>
<td>LWE</td>
<td>depth($f$)</td>
<td>[ABJ+19]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[QWW18]</td>
<td></td>
<td></td>
<td>[CCG+20]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[AJJM20]</td>
<td>R-LWE, DSPR &amp; OT</td>
<td>$L_{in}$ &amp; $L_{out}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Prior Work

<table>
<thead>
<tr>
<th>Round-Optimal Protocols</th>
<th>Semi-Honest</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td>[GS18]</td>
<td>OT</td>
<td>[CCG+20]</td>
</tr>
<tr>
<td>With Improved Comm. Compl.</td>
<td>LWE, depth(ψ)</td>
<td>This Work</td>
</tr>
<tr>
<td>[ABJ+19]</td>
<td>LWE</td>
<td></td>
</tr>
<tr>
<td>[QWW18]</td>
<td>LWE</td>
<td></td>
</tr>
<tr>
<td>[AJJM20]</td>
<td>R-LWE, DSPR &amp; OT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$L_{in}$ &amp; $L_{out}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semi-Honest</td>
<td>Malicious</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Work</td>
<td>Work</td>
</tr>
<tr>
<td>With Improved Comm. Compl.</td>
<td>[ABJ+19] [QWW18] LWE depth($f$)</td>
<td>[AJJM20] R-LWE, DSPR &amp; OT $L_{in}$, $L_{out}$</td>
</tr>
<tr>
<td></td>
<td>This Work</td>
<td>LWE depth($f$)</td>
</tr>
</tbody>
</table>
## Prior Work

<table>
<thead>
<tr>
<th></th>
<th>Semi-Honest</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Round-Optimal Protocols</strong></td>
<td>[BL18] [GS18]</td>
<td>OT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>With Improved Comm. Compl.</strong></td>
<td>[ABJ+19] [QWW18]</td>
<td>LWE</td>
</tr>
<tr>
<td></td>
<td>[AJJM20]</td>
<td>R-LWE, DSPR &amp; OT</td>
</tr>
</tbody>
</table>
### Prior Work

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-Honest</td>
<td></td>
<td></td>
<td>Malicious</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[GS18]</td>
<td></td>
<td></td>
<td>[CCG+20]</td>
<td>OT</td>
<td></td>
</tr>
<tr>
<td>With Improved Comm. Compl.</td>
<td>[ABJ+19]</td>
<td>LWE</td>
<td>This Work</td>
<td>LWE</td>
<td></td>
</tr>
<tr>
<td>[QWW18]</td>
<td>depth($f$)</td>
<td></td>
<td></td>
<td>depth($f$)</td>
<td></td>
</tr>
<tr>
<td>[AJJM20]</td>
<td>R-LWE, DSPR &amp; OT</td>
<td>$L_{in} &amp; L_{out}$</td>
<td></td>
<td>R-LWE, DSPR &amp; OT</td>
<td>$L_{in} &amp; L_{out}$</td>
</tr>
</tbody>
</table>

⇒ Start from the work of Ananth et al. and Functional Encryption Combiners
Functional Encryption [BSW11]
Functional Encryption [BSW11]
Functional Encryption [BSW11]

\[ \text{Setup} \leftarrow \text{Keygen}(\mathcal{K}, f) \]
Functional Encryption [BSW11]

\[
\text{Keygen}(\text{Setup}, f) \quad \leftarrow \quad x \leftarrow \text{Enc}(\text{Keygen}, x)
\]
Functional Encryption [BSW11]

\[
\text{Setup} \leftarrow \text{Setup} \\
\text{Keygen} \leftarrow \text{Keygen} (\text{Setup}, f) \\
x \leftarrow \text{Enc} (\text{Setup}, x)
\]
**Functional Encryption [BSW11]**

\[ f(x) = \text{Dec}(Key_f, Enc(x)) \]
(Decomposable) Functional Encryption Combiner [ABJ+19]
(Decomposable) Functional Encryption Combiner [ABJ+19]
(Decomposable) Functional Encryption Combiner [ABJ+19]
(Decomposable) Functional Encryption Combiner [ABJ+19]

\[ \begin{align*}
    &\text{Setup} \\
    &\text{Keygen}(1, f) \\
    &\text{Setup} \\
    &\text{Keygen}(2, f) \\
\end{align*} \]
(Decomposable) Functional Encryption Combiner [ABJ+19]
(Decomposable) Functional Encryption Combiner [ABJ+19]

\[ f(x_1, x_2) = \text{Dec}(\text{Comb}(\text{Keygen}(k_1, f), k_2), \text{Enc}(\text{Keygen}(k_1, f), (x_1, x_2)))) \]
(Decomposable) Functional Encryption Combiner [ABJ+19]

Succinctness: $|\alpha_{f,i}| \leq \text{depth}(f)$
Protocol of Ananth et al. [ABJ+19]
Protocol of Ananth et al. [ABJ+19]
Protocol of Ananth et al. [ABJ+19]
Protocol of Ananth et al. [ABJ+19]

\[ f_1 \leftarrow \text{Setup} \]

\[ f_{i,1} \leftarrow \text{Keygen}(f_{1,i}, f) \]

\[ f_2 \leftarrow \text{Setup} \]

\[ f_{i,2} \leftarrow \text{Keygen}(f_{2,i}, f) \]
Protocol of Ananth et al. [ABJ+19]
Protocol of Ananth et al. [ABJ+19]

\[ f,1 \leftarrow \text{Keygen}(\sigma_{1}, f) \]

\[ f,2 \leftarrow \text{Keygen}(\sigma_{2}, f) \]
Protocol of Ananth et al. [ABJ+19]

\[ f, 1 \leftarrow \text{Setup} \]
\[ f, 1 \leftarrow \text{Keygen}(1, f) \]
\[ f, 2 \leftarrow \text{Keygen}(2, f) \]

\[ (x_1, x_2) = \text{Enc}(1, (x_1, x_2), (\bullet, \bullet)) \]
Protocol of Ananth et al. [ABJ+19]

\[f(x_1, x_2) = \text{Dec}(\text{f}, (x_1, x_2))\]

\[f(x_1, x_2) = \text{Enc}(\text{f}, (x_1, x_2), ((x_1, x_2), (x_1, x_2)))\]
Protocol of Ananth et al. [ABJ+19]

\[ f(x_1, x_2) = \text{Dec}(f_j, (x_1, x_2)) \]

⇒ Replace semi-honest protocol with maliciously secure protocol
First Approach

\begin{align*}
\text{Setup} &\quad \rightarrow \quad \text{Setup} \\
\text{Keygen}(\text{Setup}, f) &\quad \rightarrow \quad \text{Keygen}(\text{Setup}, f) \\
\text{Dec}(f(x_1, x_2)) &\quad = \quad \text{Enc}(\text{Setup}, (x_1, x_2), (\text{Setup}, f)) \\
\end{align*}
First Approach

1. \( f_{i,i} \) can be generated maliciously
First Approach

1. $f_{i,i}$ can be generated maliciously
2. $(\pi, \eta)$ used for encryption can be “bad”
First Approach

1. \( f_i \) can be generated maliciously
2. \((\bullet, \circ)\) used for encryption can be “bad”
3. \( i \) can be generated arbitrarily
First Approach

1. \( f_{i,j} \) can be generated maliciously
First Approach

$$f(x_1, x_2) = \text{Dec}(f, (x_1, x_2))$$

1. $f_i$ can be generated maliciously
   → Privacy with Knowledge of Outputs
First Approach

1. $f_i$ can be generated maliciously
   → Privacy with Knowledge of Outputs
   → Can be lifted using [IKP10, PC12]
First Approach

1. $f_{i,j}$ can be generated maliciously ✓
2. $(\vdash, \bowtie)$ used for encryption can be “bad”
3. $i$ can be generated arbitrarily
First Approach

2. \((\text{key}, \text{value})\) used for encryption can be “bad”
First Approach

1. $f_{1,1} \leftarrow \text{Setup}$
2. $f_{1,1} \leftarrow \text{Keygen}(1, f)$
3. $f_{1,2} \leftarrow \text{Setup}$
4. $f_{1,2} \leftarrow \text{Keygen}(2, f)$

$f(x_1, x_2) = \text{Dec}(f, (x_1, x_2)) = \text{Enc}(1, (x_1, x_2), (\text{key}, \text{key}))$

2. $(\text{key}, \text{key})$ used for encryption can be “bad”
   → Use XOR instead of concatenation
First Approach

1. Setup

2. Keygen($f_1$, $x_1$) → $f_1$

3. $f_1$, $x_1$ → Keygen($f_2$, $x_2$)

$f(x_1, x_2) = \text{Dec}(f, (x_1, x_2))$

$= \text{Enc}(\text{Setup}, (x_1, x_2), (\text{\textcolor{red}{\textbf{\textbullet}} \textcolor{red}{\textbf{\textbullet}})))$

2. $(\textcolor{red}{\textbf{\textbullet}}, \textcolor{red}{\textbf{\textbullet}})$ used for encryption can be “bad”
   → Use XOR instead of concatenation
First Approach

1. $f_{ij}$ can be generated maliciously ✓
2. $(\mathcal{O}, \mathcal{O})$ used for encryption can be “bad” ✓
3. $i$ can be generated arbitrarily

$$f(x_1, x_2) = \text{Dec}(f_{ij}, (x_1, x_2))$$

$$f(x_1, x_2) = \text{Dec}(f_{ij}, (x_1, x_2))$$
First Approach

1. $f, 1 \leftarrow \text{Setup}$
2. $f, 1 \leftarrow \text{Keygen}(\underline{1}, f)$
3. $f(x_1, x_2) = \text{Dec}(\underline{f}, (x_1, x_2))$

3. $f(x_1, x_2)$ can be generated arbitrarily
First Approach

1. $f_{i,1} \leftarrow \text{Setup}$
2. $f_{i,1} \leftarrow \text{Keygen}(a_{1}, f)$
3. $f(x_1, x_2) = \text{Dec}(f_{i,2}, (x_1, x_2))$

3. $f_{i,1}$ can be generated arbitrarily
   $\rightarrow$ Solve as 2.
3. $i$ can be generated arbitrarily
   $\rightarrow$ Solve as 2.
First Approach

3. $i$ can be generated arbitrarily
   → Solve as 2.
First Approach

3. $x_i$ can be generated arbitrarily
   \[ \rightarrow \text{Solve as 2.} \]
First Approach

3. \( \mathcal{E}_i \) can be generated arbitrarily

\[ \rightarrow \text{Solve as 2.} \]
First Approach

3. $i$ can be generated arbitrarily
   $\rightarrow$ Solve as 2.
First Approach

3. $i_i$ can be generated arbitrarily
   $\rightarrow$ Solve as 2.
First Approach

3. $\mathcal{K}_i$ can be generated arbitrarily
   $\rightarrow$ Solve as 2.
First Approach

\[ f(x_1, x_2) = \text{Dec} \left( f_{i}, (x_i, x_2) \right) \]

3. \( f_i \) can be generated arbitrarily
   \( \rightarrow \) Solve as 2. \( \times \) \( \rightarrow \) Adds an additional round
First Approach

3. $i$ can be generated arbitrarily
   → Solve as 2. $\times$ → Adds an additional round
   → Do Coin Flipping outside of protocol
3. \( r_i \) can be generated arbitrarily
   \[ \rightarrow \text{Solve as 2.} \quad \times \rightarrow \text{Adds an additional round} \]
   \[ \rightarrow \text{Do Coin Flipping outside of protocol} \]
3. $i$ can be generated arbitrarily
   → Solve as 2. $\times$ → Adds an additional round
   → Do Coin Flipping outside of protocol
3. $i$ can be generated arbitrarily
   → Solve as 2. $\times$ → Adds an additional round
   → Do Coin Flipping outside of protocol
Final Approach

3. $i$ can be generated arbitrarily
   → Solve as 2. $\times$ → Adds an additional round
   → Do Coin Flipping outside of protocol
Final Approach

3.  can be generated arbitrarily
   → Solve as 2.  → Adds an additional round
   → Do Coin Flipping outside of protocol
3. $i$ can be generated arbitrarily
   → Solve as 2. $x$ → Adds an additional round
   → Do Coin Flipping outside of protocol
Final Approach

3. $i$ can be generated arbitrarily
   → Solve as 2.  $\times$  → Adds an additional round
   → Do Coin Flipping outside of protocol
Final Approach

3. A key $k_i$ can be generated arbitrarily
   → Solve as 2. $×$ → Adds an additional round
   → Do Coin Flipping outside of protocol
Final Approach

1. $\leftarrow \text{Setup}(\mathcal{I})$
2. $f_{i,1} \leftarrow \text{Keygen}(\mathcal{I}_{1, f})$
3. $x_{1, i} \leftarrow (\mathcal{I}_{1, f}, \mathcal{I}_{2, f})$
4. $x_{1, i} = (\mathcal{I}_{2, f}, \mathcal{I}_{1, f})$
5. $\leftarrow \text{Setup}(\mathcal{I})$
6. $f_{i,2} \leftarrow \text{Keygen}(\mathcal{I}_{2, f})$
7. $, \mathcal{I}_{1, f} = (\mathcal{I}_{1, f}, \mathcal{I}_{2, f})$

3. $i$ can be generated arbitrarily
   → Solve as 2. $x$ → Adds an additional round
   → Do Coin Flipping outside of protocol
Final Approach

3. $i$ can be generated arbitrarily
   → Solve as 2. $x$ → Adds an additional round
   → Do Coin Flipping outside of protocol
3. $c_i$ can be generated arbitrarily
   → Solve as 2. $x$ → Adds an additional round
   → Do Coin Flipping outside of protocol
3. $\iota_i$ can be generated arbitrarily
   $\rightarrow$ Solve as 2. $\times$ $\rightarrow$ Adds an additional round
   $\rightarrow$ Do Coin Flipping outside of protocol
3. $k_{i}$ can be generated arbitrarily
   → Solve as 2. $x$ → Adds an additional round
   → Do Coin Flipping outside of protocol → How to allow for honest behavior check?
3. $\sigma_i$ can be generated arbitrarily
   → Solve as 2. $\times$ → Adds an additional round
   → Do Coin Flipping outside of protocol → How to allow for honest behavior check?
   ⇒ $k$-Delayed-Input Function MPC
$k$-Delayed-Input vs. $k$-Delayed-Input Function MPC
$k$-Delayed-Input vs. $k$-Delayed-Input Function MPC

$k$-Delayed-Input MPC:

1. The input is needed in round $k$
$k$-Delayed-Input vs. $k$-Delayed-Input Function MPC

$k$-Delayed-Input MPC:

1. The input is needed in round $k$

2. but needs to be fixed before the protocol execution
$k$-Delayed-Input vs. $k$-Delayed-Input Function MPC

$k$-Delayed-Input MPC:

1. The input is needed in round $k$
2. but needs to be fixed before the protocol execution

$k$-Delayed-Input Function MPC:

1. The input is needed in round $k$
$k$-Delayed-Input vs. $k$-Delayed-Input Function MPC

$k$-Delayed-Input MPC:

1. The input is needed in round $k$

2. but needs to be fixed before the protocol execution

$k$-Delayed-Input Function MPC:

1. The input is needed in round $k$

2. and is partially decided during the protocol execution
$k$-Delayed-Input vs. $k$-Delayed-Input Function MPC

$k$-Delayed-Input MPC:

1. The input is needed in round $k$
2. but needs to be fixed before the protocol execution

$k$-Delayed-Input Function MPC:

1. The input is needed in round $k$
2. and is partially decided during the protocol execution

⇒ 2$n$-Party $k$-delayed-input MPC protocol + information-theoretic MAC
3. $\hat{r}_i$ can be generated arbitrarily
   → Solve as 2. $\times$ → Adds an additional round
   → Do Coin Flipping outside of protocol → How to allow for honest behavior check?
3. $\bullet_i$ can be generated arbitrarily

→ Solve as 2. $\times$ → Adds an additional round
→ Do Coin Flipping outside of protocol → How to allow for honest behavior check? $✓$
1. $f_i$ can be generated maliciously ✓
2. $(\mathcal{K}, \mathcal{K})$ used for encryption can be “bad” ✓
3. $i$ can be generated arbitrarily ✓
1. $f_{i,j}$ can be generated maliciously ✓
2. $(\mathcal{K},\mathcal{K})$ used for encryption can be “bad” ✓
3. $\mathcal{K}_i$ can be generated arbitrarily ✓

$\Rightarrow$ Communication Complexity: $\text{depth}(f)$
Final Approach

1. $f_{i_1}$ can be generated maliciously
2. $(x_1, x_2)$ used for encryption can be “bad”
3. $i$ can be generated arbitrarily

⇒ Communication Complexity: $\text{depth}(f)$ Can we do better?
Final Approach

1. \[ f_i \] can be generated maliciously ✓
2. \((\cdot, \cdot)\) used for encryption can be “bad” ✓
3. \[ i \] can be generated arbitrarily ✓

⇒ Communication Complexity: \(\text{depth}(f)\) Can we do better? Yes!
Multi-Key Fully Homomorphic Encryption [LTV12]
Multi-Key Fully Homomorphic Encryption [LTV12]

\[(\mathcal{K}_1, \mathcal{K}_1) \leftarrow \text{Setup}\]

\[(\mathcal{K}_2, \mathcal{K}_2) \leftarrow \text{Setup}\]
Multi-Key Fully Homomorphic Encryption [LTV12]

\[ (k_1, c_1) \leftarrow \text{Setup} \]
\[ x_1 \leftarrow \text{Enc}(k_1, x_1) \]

\[ (k_2, c_2) \leftarrow \text{Setup} \]
\[ x_2 \leftarrow \text{Enc}(k_2, x_2) \]
Multi-Key Fully Homomorphic Encryption [LTV12]

\[(1, 2) = (1, 1) \leftarrow \text{Setup} \]
\[x_1 \leftarrow \text{Enc}(1, x_1)\]
\[(2, 2) \leftarrow \text{Setup} \]
\[x_2 \leftarrow \text{Enc}(2, x_2)\]

\[= (1, 2)\]
Multi-Key Fully Homomorphic Encryption [LTV12]
Multi-Key Fully Homomorphic Encryption [LTV12]

\[ f(x_1, x_2) = \text{Eval}(f, x_1, x_2) \]

\[ f(x_1, x_2) = \text{Dec}(\text{Eval}(f, x_1, x_2)) \]
Multi-Key Fully Homomorphic Encryption [LTV12]

\[ f(x_1, x_2) = \text{Eval}(f, x_1, x_2) \]

Compactness: \(|f(x_1, x_2)|\) independent of \(f\)
Round-Optimal and Communication-Efficient MPC
Round-Optimal and Communication-Efficient MPC

$(σ_1, ρ_1) \leftarrow \text{Setup}$

$(σ_2, ρ_2) \leftarrow \text{Setup}$
Round-Optimal and Communication-Efficient MPC

\[(x_1, x_1) \leftarrow \text{Setup} \]
\[x_1 \leftarrow \text{Enc}(x_1, x_1)\]

\[(x_2, x_2) \leftarrow \text{Setup} \]
\[x_2 \leftarrow \text{Enc}(x_2, x_2)\]
Round-Optimal and Communication-Efficient MPC

1. Setup

2. Enc($x_1$, $x_1$)

3. $x_1 \leftarrow$ Enc($x_1$, $x_1$)

4. Setup

5. Enc($x_2$, $x_2$)

6. $x_2 \leftarrow$ Enc($x_2$, $x_2$)
Round-Optimal and Communication-Efficient MPC

\((1, 1) \leftarrow \text{Setup}\)
\(x_1 \leftarrow \text{Enc}(1, x_1)\)
\(f(x_1, x_2) \leftarrow \text{Eval}(f, x_1, x_2)\)
\(\)
Round-Optimal and Communication-Efficient MPC

Setup

Encryption

Evaluation

Setup

Encryption

Evaluation
Round-Optimal and Communication-Efficient MPC

\[ \left( x_1, x_1 \right) \leftarrow \text{Setup} \]

\[ x_1 \leftarrow \text{Enc}(x_1, x_1) \]

\[ f(x_1, x_2) \leftarrow \text{Eval}(f, x_1, x_2) \]

\[ \text{Check w.r.t. } f(x_1, x_2) \]

\[ \left( x_2, x_2 \right) \leftarrow \text{Setup} \]

\[ x_2 \leftarrow \text{Enc}(x_2, x_2) \]

\[ f(x_1, x_2) \leftarrow \text{Eval}(f, x_1, x_2) \]

\[ \text{Check w.r.t. } f(x_1, x_2) \]
Round-Optimal and Communication-Efficient MPC

(1, 1) ← Setup
x₁ ← Enc(₁, x₁)

(1, x₁) ← Ḷ𝖾𝗍𝗎𝗉 ← Ḷ𝗇𝖼(₁, x₁)

x₁ ← Ḷ𝖾𝗍𝗎𝗉 ← Ḷ𝗇𝖼(₁, x₁)

(2, 2) ← Setup
x₂ ← Enc(₂, x₂)

(2, x₂) ← Ḷ𝖾𝗍𝗎𝗉 ← Ḷ𝗇𝖼(₂, x₂)

x₂ ← Ḷ𝖾𝗍𝗎𝗉 ← Ḷ𝗇𝖼(₂, x₂)

f(x₁, x₂) ← Eval(₁, f, x₁, x₂)

f(x₁, x₂), (, ) ← Ḷ𝗉𝖺𝗅(₁, f, x₁, x₂)

f(x₁, x₂), (, ) ← Ḷ𝗉𝖺𝗅(₁, f, x₁, x₂)

Check w.r.t. & w.r.t.

f(x₁, x₂) = Dec(₂, f(x₁, x₂))
Round-Optimal and Communication-Efficient MPC

Check w.r.t. $f(x_1, x_2)$ = Dec($f(x_1, x_2)$)

$\Rightarrow$ Communication Complexity: $L_{\text{in}}$ & $L_{\text{out}}$ independent of $f$
Conclusion
Conclusion

- Round-Optimal and Communication-Efficient Multiparty Computation
Conclusion

- Round-Optimal and Communication-Efficient Multiparty Computation
  - Protocol with Communication Complexity depth($f$)
    based on Functional Encryption Combiners
Conclusion

- Round-Optimal and Communication-Efficient Multiparty Computation
  - Protocol with Communication Complexity depth$(f)$ based on Functional Encryption Combiners
  - Protocol with Communication Complexity $L_{\text{in}}$ & $L_{\text{out}}$ based on Multi-Key Fully Homomorphic Encryption
Conclusion

- Round-Optimal and Communication-Efficient Multiparty Computation
  - Protocol with Communication Complexity $\text{depth}(f)$ based on Functional Encryption Combiners
  - Protocol with Communication Complexity $L_{\text{in}}$ & $L_{\text{out}}$ based on Multi-Key Fully Homomorphic Encryption
- $k$-Delayed-Input Function MPC
Conclusion

● Round-Optimal and Communication-Efficient Multiparty Computation

○ Protocol with Communication Complexity $\text{depth}(f)$ based on Functional Encryption Combiners

○ Protocol with Communication Complexity $L_{\text{in}} \& L_{\text{out}}$ based on Multi-Key Fully Homomorphic Encryption

● $k$-Delayed-Input Function MPC

Thank You!