Efficient secure two-party computation secure against active adversary using Yao's Garbled Circuit and GMW paradigm

Mohammad Hossein Faghihi Sereshgi
University Of Rochester

Collaborators: Jackson Abascal, Carmit Hazay, Yuval Ishai, Muthuramakrishnan Venkitasubramaniam
Outline

• What is secure Multi-party Computation
• Yao’s Garbled Circuit
• Protocol with Active security
• Proof of Security
• Results
Secure Multi-Party Computation

• $n$ parties want to compute $F(x_1, x_2, \ldots, x_n)$
  • Keep the inputs private
    • No one learns anything more than the output of the function

$x_1$ $x_2$

$F(x_1, x_2, \ldots, x_n)$

$x_3$ $\ldots$ $x_n$
Secure Multi-Party Computation

- Trusted Third Party
  - Receives the inputs and returns the output

\[ F(x_1, x_2, \ldots, x_n) \]
Secure Multi-Party Computation

• It is almost impossible to find an entity trusted by everyone
Secure Multi-Party Computation

• Use a protocol that does not need a TTP.

\[ F(x_1, x_2, \ldots, x_n) \]
Secure Multi-Party Computation

• Secure Two-Party Computation
Adversary

• Two types of adversary
  • Semi-Honest (Passive, Honest-but-curious)
    • Follows the protocol
    • Investigates the communications
  • Malicious (Active, Byzantine)
    • Deviates from the protocol
    • Sends bogus messages or goes offline
    • Adversary wants to violate correctness of result and privacy
Yao’s Garbled Circuit [Yao96]

• One of the first protocol for 2PC

• Passive security

• Assumption:
  • Oblivious Transfer
Oblivious Transfer

Sender

\[ S_0, S_1 \]

\[ b \]

\[ S_b \]

Receiver

Sender Does not learn \( b \)

Receiver Does not learn \( S_{1-b} \)
Yao’s GC

Garbler’s message containing $\mathcal{C}$ and $\mathcal{X}_G$

Evaluator’s message to OT

Evaluators encoded input $\bar{\mathcal{X}}_E$

Garbler’s message to OT

Garbler’s message containing $\bar{\mathcal{C}}$ and $\bar{\mathcal{X}}_G$
Yao’s GC

• Consider a circuit $C$ that computes the function $F$
Yao’s GC

\[ k^i_b \in \{0,1\}^\kappa \]
Yao’s GC

- Garbling AND gate

\[
\begin{array}{cccc}
0,1 & \text{AND} & 0,1 \\
0,1 & & & \\
\end{array}
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccc}
k_0^0, k_1^0 & \text{AND} & k_0^8, k_1^8 \\
k_0^1, k_1^1 & & & \\
\end{array}
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_0^0)</td>
<td>(k_0^1)</td>
<td>(\text{Enc}<em>{k_0^0}(\text{Enc}</em>{k_0^1}(k_0^8)))</td>
</tr>
<tr>
<td>(k_0^0)</td>
<td>(k_1^1)</td>
<td>(\text{Enc}<em>{k_0^0}(\text{Enc}</em>{k_1^1}(k_0^8)))</td>
</tr>
<tr>
<td>(k_1^0)</td>
<td>(k_0^1)</td>
<td>(\text{Enc}<em>{k_1^0}(\text{Enc}</em>{k_0^1}(k_0^8)))</td>
</tr>
<tr>
<td>(k_1^0)</td>
<td>(k_1^1)</td>
<td>(\text{Enc}<em>{k_1^0}(\text{Enc}</em>{k_1^1}(k_1^8)))</td>
</tr>
</tbody>
</table>
Yao’s GC

• Garbler sends the encoded truth table and his encoded input
• For Evaluator’s input, they use OT

Garbler sends the encoded truth table and his encoded input. For Evaluator’s input, they use OT. In the OT protocol, the Garbler sends $k_0^1, k_1^1$ to the OT protocol, and the Evaluator sends $b$ to the OT protocol. The Receiver does not learn $k_1^1$, and the Sender does not learn $b$. The Evaluator does not learn $k_{1-b}^1$. 
Yao’s GC

• Evaluator decrypts all possible outputs
  • Only one of them will be valid

• Assume Garbler’s input is 1 and Evaluator’s input is 0

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>Dec_{k_1} \left( Dec_{k_0} (\cdots) \right)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k_0^0</td>
<td>k_1^1</td>
<td>Enc_{k_0} \left( Enc_{k_1} \left( k_0^8 \right) \right)</td>
<td>Invalid!</td>
</tr>
<tr>
<td>k_0^0</td>
<td>k_1^1</td>
<td>Enc_{k_0} \left( Enc_{k_1} \left( k_0^8 \right) \right)</td>
<td>Dec_{k_1} \left( Dec_{k_0} (\cdots) \right)</td>
</tr>
<tr>
<td>k_1^0</td>
<td>k_1^1</td>
<td>Enc_{k_1} \left( Enc_{k_0} \left( k_1^8 \right) \right)</td>
<td>Dec_{k_1} \left( Dec_{k_0} (\cdots) \right)</td>
</tr>
<tr>
<td>k_1^0</td>
<td>k_1^1</td>
<td>Enc_{k_1} \left( Enc_{k_1} \left( k_1^8 \right) \right)</td>
<td>Dec_{k_1} \left( Dec_{k_0} (\cdots) \right)</td>
</tr>
</tbody>
</table>
Yao’s GC

- **Complexities:**
  - Communication: $O(\kappa|C|)$ bits
  - $O(|C|)$ PRG invocation
  - $n$ Oblivious Transfer on pairs of $\kappa$-bit strings
    - $n$: length of Evaluator’s input
Yao’s GC

- Secure against passive (honest-but-curious) adversary
- In the OT-hybrid, the protocol is secure against actively corrupted Evaluator
- However, an actively corrupted Garbler can attack the protocol!
Yao’s GC

• Theoretical solution
  • GMW Paradigm [GMW87]: Attach a zero-knowledge proof (ZK) with every message
  • Not considered practical!

• Concretely efficient solutions:
  • Cut-and-Choose [LP07,...]
  • Authenticated Garbling [IKOPS11,WRK17,YWZ19]

This Work: GMW is practical!
Yao’s GC

• Timeline of some of the works on 2PC
Zero-knowledge proof

• Prover $P$ has witness $w$ that $x \in L$ and wants to convince $V$ that $x \in L$
• Soundness: if $x \notin L$, a cheating $P^*$ cannot convince $V$
• Zero Knowledge: The protocol reveals nothing more than $x \in L$
Active Security

GMW Paradigm [GMW87]

• ZKP + passive security = Active security
  • Costly
Comparison

• Asymptotic Complexity

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Func-ind (Comm./Comp.)</th>
<th>Func-dep (Comm./Comp.)</th>
<th>Online (Comm.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Yao86]</td>
<td>$O(</td>
<td>C</td>
<td>k)$</td>
</tr>
<tr>
<td>[HIV17]</td>
<td>$O(</td>
<td>C</td>
<td>k)$ (Input dependent)</td>
</tr>
<tr>
<td>Authenticated garble[WRK17]</td>
<td>$O\left(\frac{</td>
<td>C</td>
<td>\rho k}{\log \tau + \log</td>
</tr>
<tr>
<td>[IPS08] in Authenticated garble[WRK17]</td>
<td>$O(</td>
<td>C</td>
<td>k)$</td>
</tr>
<tr>
<td>[AFHIV20]</td>
<td>$O\left(</td>
<td>C</td>
<td>k + \sqrt{</td>
</tr>
</tbody>
</table>

$k$ Computational security parameter  
$\rho$ Statistical security parameter  
$\tau$ Number of protocol executions in the amortized setting
Features of the protocol

• Boolean operations
  • Based on Yao’s GC

• Secure against active Garbler using ZKP
  • Uses Ligero [AHIV17]

• Offline-Online phase

• Offline Phase is non-interactive
  • The two parties do not need to know each other

• Online phase needs only one round
Yao’s GC+ZKP

• Garbler proves that:
  • The GC is constructed correctly
  • The Garbler's input is consistent with the GC
  • The Evaluator's encoded input is consistent with the GC

• First Variant: Non-black-box in PRG but black-box in OT
• How? Certified OT [IKOPS11,HIV17]
Certified OT

- COT is parameterized with an NP-relation $R$
- The receiver will receive the output only if the relation is true

We modularly show how to realize COT using OT in a black-box way
Certified OT

Sender

\[(s^0_j, s^1_j)_{j \in [n]}\]

Commit \(\left\{ s^b_j \right\}_{j \in [n], b \in \{0,1\}}\)

\(\mathcal{H}\)

\(\left\{ \mathcal{H} \left( s^b_j \right) \right\}_{j \in [m], b \in \{0,1\}}\)

ZKP

OT

\(b^1, \ldots, b^n\)

\(\left( s^b_j \right)_{j \in [n]}\)

Receiver

Sample a MAC key \(\mathcal{H}\)

• ZKP shows that
  • NP-relation R on sender’s input is satisfied
  • The MAC values are computed correctly
  • Can compress rounds using known (Fiat-Shamir’s) heuristic
Certified OT-Input-Value Disjunction (IVD)

\[ P = (v^1 \lor v^2 \lor \ldots \lor v^n) \]

\[ R((x, s^0_1, s^1_1, \ldots, s^0_n, s^1_n), w) \]
Certified OT- IVD

- Encode the receiver’s input in order to deal with the 1bit leakage [LP07,IKOPS11,SS13]
Proof of Security Using Simulator

• Probabilistic Polynomial-Time Turing Machine
• Generates (simulates) the view of the adversary
  • View: \{x, r, m_0, m_1, ... \}
  • Given adversary’s input and the output
Proof of Security Using Simulator

• Real World
Proof of Security Using Simulator

• Ideal World
Proof of Security Using Simulator

• Ideal World

Sender \rightarrow COT-IVD \rightarrow Receiver
Proof of Security Using Simulator
Proof of Security Using Simulator

Sender

COT-IVD

Simulator

Receiver
Certified OT-IVD: Proof of Security

Sender
(\(s_j^0, s_j^1\) \(j \in [n]\))
Commit \(\left(\{s_j^b\} \middle| j \in [n], b \in \{0,1\}\right)\)
\(\mathcal{H}\)
\(\{\mathcal{H} \left( s_j^b \right) \} \middle| j \in [m], b \in \{0,1\}\)
ZKP

Simulator
\(b^1, \ldots, b^n\)
OT
Sample a MAC key \(\mathcal{H}\)
Receiver
Certified OT-IVD: Proof of Security

Simulator

Sender

(b_{1},...b_{n})

OT

(s_{j}^{0},s_{j}^{1})_{j \in [n]}

Commit \((s_{j}^{b_{j}})_{j \in [n]}\)

(\{s_{j}^{b}\}_{j \in [n], b \in \{0,1\}})\)

H

(\{H(s_{j}^{b})\}_{j \in [m], b \in \{0,1\}})

ZKP

Sample a MAC key H

Receiver
Offline-Online setting

- The GC Proof is input independent
  - Can be done offline without interaction (Silent preprocessing)
  - The Garbler can make the GC and the ZKP available on internet.
- The Evaluator’s message for OT protocol does not need Interaction.
  - The Evaluator can make it available on internet before protocol starts.
Offline-Online setting

We split the protocol in Offline phase and Online phase

• Offline phase
  • Garbler publishes the GC and its proof of correctness
  • Evaluator publishes the first message of the OT protocol

• Online phase
  • Garbler sends the response to OT
  • Garbler sends a proof that the labels transmitted are consistent with the GC
Offline-Online setting

• Split the zero-knowledge proofs into two parts:
  • $ZK_{\text{off}}$
    • GC is constructed correctly
  • $ZK_{\text{on}}$
    • Inputs to OT functionality are consistent with the GC

• Need a commit-and-prove system where we can give multiple proofs on committed values
  • Instantiate using MPC-in-the-head paradigm [IKOS07]
  • Design a concretely efficient variant with sublinear communication complexity (using a variant of Ligero [AHIV17])
Variants of the protocol

• Variant 1: Certified OT
  • Implementation!
  • Most communication efficient 2PC to date
  • Competitive computational complexity

• Variant 2: OT (Non-black-box on OT and PRG)
  • Larger ZKP in the online phase. Competitive for large input sizes
  • Reusable (Non-Interactive Secure Computation) NISC!
Results - Offline

![Graphs showing communication and time vs. AND gates for different protocols.](image)

- AES10
- AES2
- WRK Offline
- Garbled Circuit size

![Graphs showing communication and time vs. AND gates for different protocols.](image)
Results - Online

[Graphs showing the relationship between online proof length and input size, as well as the time taken for different input sizes for Receiver and Sender.]
Thank You