CS 65500
Advanced Cryptography

Lecture 6: Semi-Honest GMW - II

Instructor: Aarushi Goel

Spring 2025

Agenda

- → Recall the semi-honest GMW protocol
- → Security Proof.

Reminder: HW2 will be released today

Secure Two-Party Computation of *General* Functions

Alice

Bob

Input: Alice: $x_1, \ldots, x_m \in \{0,1\}^m$ Bob: $y_1, \ldots, y_m \in \{0,1\}^m$

Function: $f : \{0,1\}^{2m} \longrightarrow \{0,1\}^{\ell}$

Output: $f(x_1, \ldots, x_m, y_1, \ldots, y_m) = z_1, \ldots, z_\ell$

Function Representation

Function $f:\{0,1\}^{2m} \to \{0,1\}^{l}$ can be represented as a Boolean circuit:

Input wires: $x_1, ---, x_m, y_1, --, y_m$
Output wires: $z_1, ---, z_l$

Gates: Since NAND gates are complete, we will assume that the circuit only comprises of AND and NOT gates.

GMW Protocol : Input Sharing

Alice

Bob

Inputs : $x_1, \ldots, x_m$ $\qquad$ $y_1, \ldots, y_m$

$\forall i \in [m]$:

$\text{Share}(x_i) \rightarrow x_i^A, x_i^B$

$\forall i \in [m]$:

$\text{Share}(y_i) \rightarrow y_i^A, y_i^B$

$$\xrightarrow{\quad x_1^B, \ldots, x_m^B \quad}$$

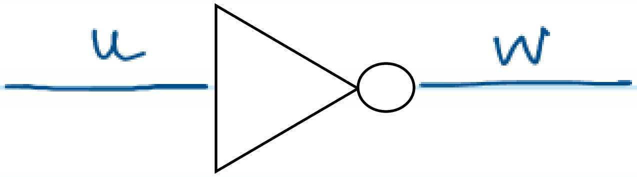
$$\xleftarrow{\quad y_1^A, \ldots, y_m^B \quad}$$

GMW Protocol : Circuit Evaluation

Alice

Bob

NOT gate



Alice holds u^A

compute $w^A = u^A \oplus 1$

Bob holds u^B

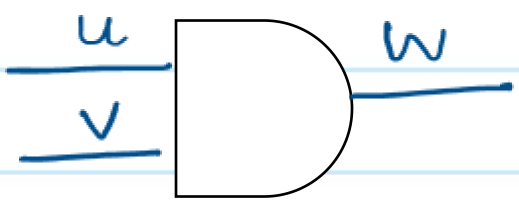
compute $w^B = u^B$

Notice that $w^A \oplus w^B = u^A \oplus 1 \oplus u^B = \overline{u}$

$\Rightarrow$ invariant is maintained !!

GMW Protocol : Circuit Evaluation

AND gate



- Alice holds u^A, v^A
- Sample $r \leftarrow \{0,1\}$ and use the following inputs to π_{OT}

$$a_{00} = r \oplus ((u^A \cdot 0) \oplus (v^A \cdot 0))$$
$$a_{01} = r \oplus ((u^A \cdot 1) \oplus (v^A \cdot 0))$$
$$a_{10} = r \oplus ((u^A \cdot 0) \oplus (v^A \cdot 1))$$
$$a_{11} = r \oplus ((u^A \cdot 1) \oplus (v^A \cdot 1))$$

- Bob holds u^B, v^B
- use (u^B, v^B) as input the the OT protocol.

- $w^A = u^A \cdot v^A \oplus r$
- $w^B = u^B \cdot v^B \oplus a_{u^B v^B}$

Invariant is maintained!

GMW Protocol: Output Reconstruction

Alice

Bob

For all output wires: $z_1, \ldots, z_\ell$:

Alice holds $z_1^A, \ldots, z_\ell^A$

Bob holds $z_1^B, \ldots, z_\ell^B$

$$\text{Alice} \xrightarrow{z_1^A, \ldots, z_\ell^A} \text{Bob}$$

$$\text{Alice} \xleftarrow{z_1^B, \ldots, z_\ell^B} \text{Bob}$$

Alice: $\forall i \in [\ell]$

$z_i = z_i^A \oplus z_i^B$

Bob: $\forall i \in [\ell]$

$z_i = z_i^B \oplus z_i^A$

Security of GMW protocol

What do we want to Prove?

GMW is a semi-honest secure two-party computation protocol for $f:\{0,1\}^{2m} \to \{0,1\}^{\ell}$

$\exists$ a pair of n.u. PPT simulators S_A, S_B, such that $\forall k \in [n]$, and all inputs $x_1, \ldots, x_m, y_1, \ldots, y_m \in \{0,1\}^{2m}$:

$$\{S_A(x_1, \ldots, x_m, z_1, \ldots, z_\ell), z_1, \ldots, z_\ell\} \approx_c \{\mathrm{View}_A(\pi), \mathrm{Out}_B(\pi)\}$$

$$\{S_B(y_1, \ldots y_m, z_1, \ldots, z_\ell), z_1, \ldots, z_\ell\} \approx_c \{\mathrm{View}_B(\pi), \mathrm{Out}_A(\pi)\}$$

Security of GMW protocol

What do we already know?

1. π^{OT} is a semi-honest secure 1-out-of-4 Oblivious transfer protocol.

2. Additive secret sharing is a perfectly secure (2,2) secret-sharing scheme.

Security of GMW Protocol

What do we already know?

1. Semi-honest Security of Π_{OT}:

$\exists$ a pair of n.u. PPT simulators S_A^{OT}, S_B^{OT}, such that $\forall K \in [n]$, $\forall$ inputs $a_1, a_2, a_3, a_4, b_1, b_2 \in \{0,1\}^6$:

$$\left\{ S_A^{OT}(a_1 a_2 a_3 a_4, \perp), a_{b_1 b_2} \right\} \approx_c \left\{ \mathrm{View}_A^{OT}(\Pi_{OT}), \mathrm{Out}_B^{OT}(\Pi_{OT}) \right\}$$

$$\left\{ S_B^{OT}(b_1 b_2, a_{b_1,b_2}), \perp \right\} \approx_c \left\{ \mathrm{View}_B^{OT}(\Pi_{OT}), \mathrm{Out}_A^{OT}(\Pi_{OT}) \right\}$$

Security of GMW Protocol

What do we already know?

2. Perfect Security of Additive Secret Sharing

$\forall\, s, s' \in \{0,1\}^2$ and for each $p \in \{A, B\}$, the following distributions are identical:

$$\{ s_p \,;\; (s_A, s_B) \leftarrow \text{Share}(s) \} \quad \text{and}$$

$$\{ s'_p \,;\; (s'_A, s'_B) \leftarrow \text{Share}(s') \}.$$

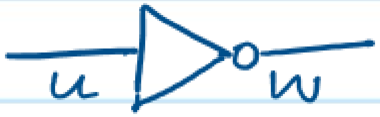
Security Proof:

Simulator $S_A(x_1, \dots, x_m, z_1, \dots, z_\ell)$

1. Input Sharing: * $\forall i \in [m]$, compute $x_i^A, x_i^B \leftarrow \mathrm{Share}(x_i)$

   * $\forall i \in [m]$, sample $y_i^A \xleftarrow{\$} \{0,1\}$

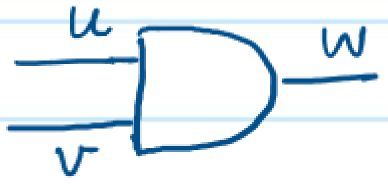
2. Circuit Evaluation: for each gate in the circuit:

   * if it is a NOT gate 

     compute $w^A = u^A \oplus 1$

Security Proof:

* if it is an AND gate



- Sample $r \xleftarrow{\$} \{0,1\}$ and compute

$$a_{00} = r \oplus ((u^A \cdot 0) \oplus (v^A \cdot 0))$$
$$a_{01} = r \oplus ((u^A \cdot 1) \oplus (v^A \cdot 0))$$
$$a_{10} = r \oplus ((u^A \cdot 0) \oplus (v^A \cdot 1))$$
$$a_{11} = r \oplus ((u^A \cdot 1) \oplus (v^A \cdot 1))$$

- Run $S_A^{OT}(a_{00}, a_{01}, a_{10}, a_{11}, \perp)$
- Compute $w^A = u^A \cdot v^A \oplus r$

3. Output Reconstruction: for each outwire z_i ($\forall i \in [l]$) compute $z_i^B = z_i \oplus z_i^A$

Output $z_1, \ldots, z_l$, and terminate.

Security Proof:

Claim: The following two distributions are computationally indistinguishable:

$$\{S_A((x_1, \ldots, x_m), (z_1, \ldots, z_\ell)), z_1, \ldots, z_\ell\} \text{ and}$$

$$\{View_A(\pi), Out_B(\pi)\}$$

Proof Idea: The differences between S_A and real execution:

1) How y_i^A is computed
2) Using S_A^{OT} instead of π^{OT} → semi-honest security of π^{OT}
3) How z_i^B is computed.

Perfect security of additive secret sharing

Security Proof:

H_1 $\{ \mathrm{View}_A(\pi), \mathrm{Out}_B(\pi) \}$

H_2 Distributed similarly to H_1, except that $\forall i \in [\ell]$
$z_i^B = z_i \oplus z_i^A$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND gate, use S_A^{OT} instead of π^{OT} to simulate Alice's view in the OT protocol.

⋮

$H_{3,G}$ Switch to S_A^{OT} instead of π^{OT} for the last AND gate.

H_4 $\{ S_A((x_1, \ldots x_m), (z_1, \ldots, z_\ell)), z_1, \ldots, z_\ell \}$

Security Proof:

H_1 $\{ View_A(\pi), Out_B(\pi) \}$

H_1 & H_2 are identically distributed

H_2 Distributed similarly to H_1, except that $\forall i \in [\ell]$
$z_i^B = z_i \oplus z_i^A$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND gate, use S_A^{OT} instead of π^{OT} to simulate Alice's view in the OT protocol.

⋮

$H_{3,G}$ Switch to S_A^{OT} instead of π^{OT} for the last AND gate.

H_4 $\{ S_A((x_1, \ldots, x_m), (z_1, \ldots, z_\ell)), z_1, \ldots, z_\ell \}$

Security Proof:

H_1 $\{ View_A(\pi), Out_B(\pi) \}$

H_2 Distributed similarly to H_1, except that $\forall i \in [\ell]$
$z_i^B = z_i \oplus z_i^A$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND gate, use S_A^{OT} instead of π^{OT} to simulate Alice's view in the OT protocol.

⋮

$H_{3,G}$ Switch to S_A^{OT} instead of π^{OT} for the last AND gate.

H_4 $\{ S_A((x_1, \ldots, x_m), (z_1, \ldots, z_\ell)), z_1, \ldots, z_\ell \}$

Security Proof:

We want to show that $\forall x_1, \ldots x_m, y_1, \ldots, y_m \in \{0,1\}^{2m}$, the following distributions are computationally indistinguishable:

H_2 Distributed similarly to the real execution, except $\forall i \in [\ell]$ $z_i^B = z_i \oplus z_i^A$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND gate, use S_A^{OT} instead of π^{OT} to simulate Alice's view in the OT protocol.

What does the security game for this indistinguishability look like?

Security Proof:

Let us assume for the sake of contradiction that $\exists$ adv A, who can distinguish between H_2 & H_3 with non-neg advantage ν. We will use this adversary to design another adv B, who can break semi-honest security of π^{OT}.

Security game for security of π^{OT} against semi-honest Alice.

 Ch

 Adv

$\xleftarrow{a_{00}, a_{01}, a_{10}, a_{11}, b_0, b_1}$

$\alpha \xleftarrow{\$} \{0,1\}$

$\xrightarrow{X}$

if $\alpha = 0$:

$X \leftarrow \{View_A(\pi^{OT}), Out_B(\pi^{OT})\}$

else:

$X \leftarrow \{Sim_A^{OT}(a_{00}, a_{01}, a_{10}, a_{11}, \perp), a_{b_0 b_1}\}$

$\xleftarrow{\alpha'}$

adv wins if $\alpha' = \alpha$

Proof by Reduction

Ch  $\alpha \xleftarrow{\$} \{0,1\}$

if $\alpha = 0$:

$X \leftarrow \{View_A(\pi^{OT}), Out_B(\pi^{OT})\}$

else:

$X \leftarrow \{Sim_A^{OT}(a_{00}, a_{01}, a_{10}, a_{11}, \perp), a_{b_0 b_1}\}$

$a_{00}, a_{01}, a_{10}, a_{11},$
u_B, v_B

X

α''

B 

- Compute input sharing phase honestly
- Prepare $a_{00}, a_{01}, a_{10}, a_{11}$ for the first AND gate
- Compute remaining AND & NOT gates honestly
- $f(x_1, \dots x_m, y_1, \dots y_m) = z_1, \dots z_u$

$\forall j \in [u]: z_j^B = z_i^A \oplus z_i$

$\{x_i, y_i\}_{i \in [m]}$

The entire transcript

α'

A

Security Proof:

H_1 $\{ \text{View}_A(\pi), \text{Out}_B(\pi) \}$

H_2 Distributed similarly to H_1, except that $\forall i \in [l]$
$z_i^B = z_i \oplus z_i^A$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND gate, use S_A^{OT} instead of π^{OT} to simulate Alice's view in the OT protocol.

⋮

$H_{3,G}$ Switch to S_A^{OT} instead of π^{OT} for the last AND gate.

H_4 $\{ S_A((x_1, \ldots x_m), (z_1, \ldots, z_l)), z_1, \ldots, z_l \}$

Security Proof:

H_1 $\{\text{View}_A(\pi), \text{Out}_B(\pi)\}$

H_2 Distributed similarly to H_1, except that $\forall i \in [l]$
$z_i^B = z_i \oplus z_i^A$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND gate, use S_A^{OT} instead of π^{OT} to simulate Alice's view in the OT protocol.

⋮

$H_{3,G}$ Switch to S_A^{OT} instead of π^{OT} for the last AND gate.

H_4 $\{S_A((x_1, \ldots x_m), (z_1, \ldots, z_l)), z_1, \ldots, z_l\}$

$H_{3,G}$ & H_4 are identically distributed.

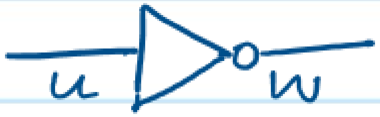
Security Proof:

Simulator $S_B(y_1, ---, y_m, z_1, ---, z_\ell)$

1. Input Sharing: * $\forall i \in [m]$, compute $y_i^A, y_i^B \leftarrow \text{Share}(y_i)$

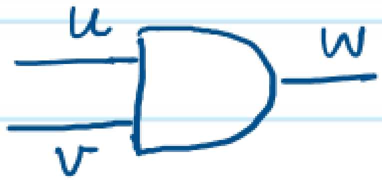
   * $\forall i \in [m]$, sample $y_i^B \xleftarrow{\$} \{0,1\}$

2. Circuit Evaluation: for each gate in the circuit:

   * if it is a NOT gate
   

   compute $w^B = u^B$

Security Proof:

* if it is an AND gate 

- Sample $s \xleftarrow{\$} \{0,1\}$
- Run $S_B^{OT}((u^B, v^B), s$
- Compute $w^B = u^B \cdot v^B \oplus s$

3. Output Reconstruction: for each outwire z_i $(\forall i \in [l])$

$$z_i^A = z_i \oplus z_i^B$$

Output $z_1, ---, z_l$ and terminate.

Security Proof:

Claim: The following two distributions are computationally indistinguishable:

$$\{S_B((y_1, \ldots, y_m), (z_1, \ldots, z_\ell)), z_1, \ldots, z_\ell\} \text{ and}$$

$$\{\mathrm{View}_B(\pi), \mathrm{Out}_A(\pi)\}$$

Proof Idea: The differences between S_B and real execution:

1) How y_i^A is computed
2) How AND gates are evaluated → semi-honest security of π^{OT}
3) How z_i^B is computed.

Perfect security of additive secret sharing

Security Proof:

H_1 $\{ View_B(\pi), Out_A(\pi) \}$

H_2 Distributed similarly to H_1, except that $\forall i \in [\ell]$ $z_i^A = z_i \oplus z_i^B$

$H_{3,1}$ Distributed similarly to H_2, except that for the first AND use $S_B^{OT}(u_B, v_B, a_{u_B v_B})$ instead of π^{OT} to simulate Bob's view in the OT protocol.

⋮

$H_{3,G}$ Switch to S_B^{OT} instead of π^{OT} for the last AND gate.

$H_{4,G}$ For the last AND gate sample $s \xleftarrow{\$} \{0,1\}$ & set $a_{u_B, v_B} = s$.

⋮

$H_{4,1}$ For the first AND gate sample $s \xleftarrow{\$} \{0,1\}$ & set $a_{u_B v_B} = s$

H_5 $\{ S_A((x_1, \ldots x_m), (z_1, \ldots, z_\ell)), z_1, \ldots, z_\ell \}$

Security Proof:

Can we change the order of hybrids?

Think about what will happen if instead of $H_{4,G}$, we have $H_{4,1}$ after $H_{3,G}$?

Exercise: Use proofs by reduction to argue indistinguishability between these hybrids.