CS 65500
Advanced Cryptography

Lecture 11: Semi-Honest BGW Protocol

Instructor: Aarushi Goel

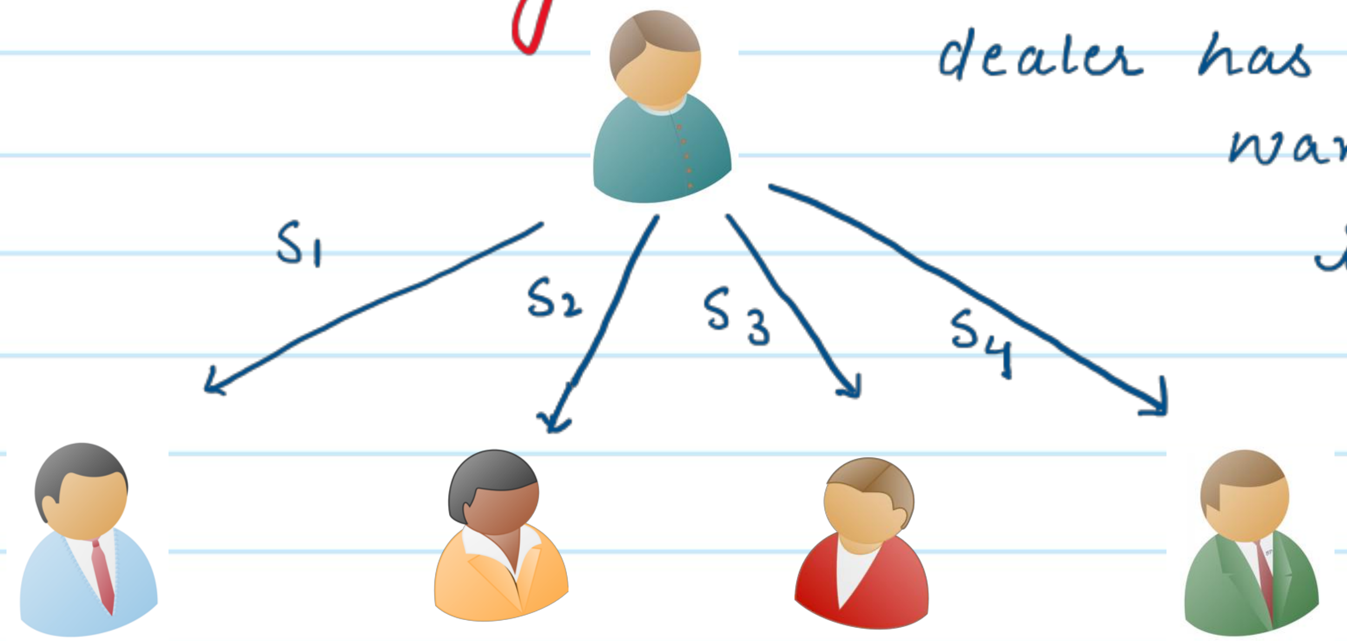
Spring 2025

Agenda

→ Semi-honest n-party secure computation protocol where $t < n/2$. (BGW Protocol)

* Construction
* Security

Secret Sharing (t, n)

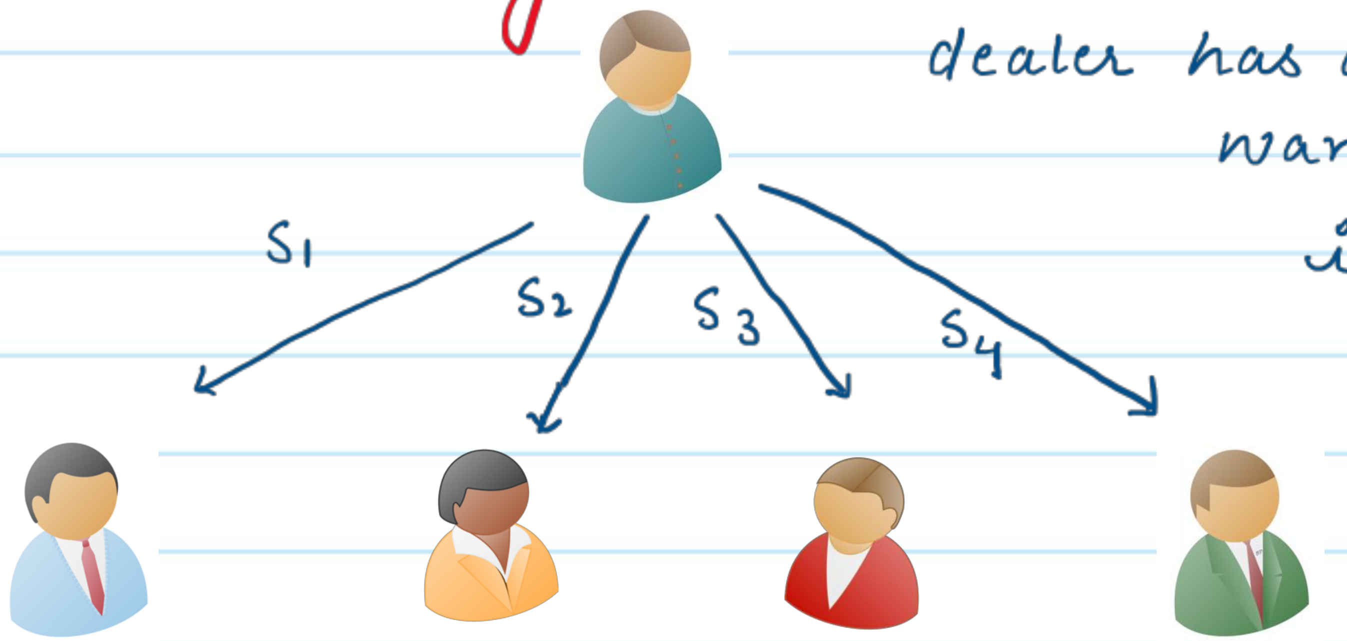


dealer has a secret s and wants to divide it into n parts/shares

Correctness: Any subset of $t+1$ shares can be combined to reconstruct the secret s.

Security: Any subset of $\leq t$ shares reveal no information about the secret s.

Secret Sharing (t, n)



dealer has a secret s and wants to divide it into n parts/shares

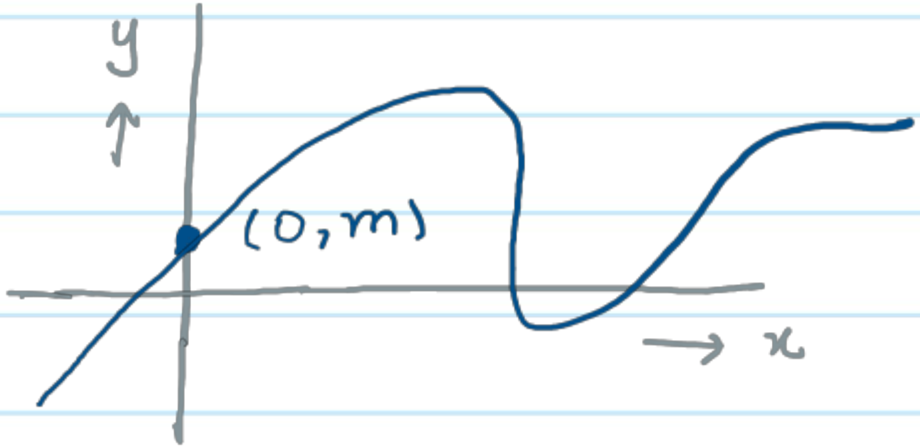
Notation: We will use $[s]_t$ to denote that a secret s has been shared using a (t,n) threshold secret sharing scheme.

Construction: (t, n) Threshold Secret Sharing
(Shamir Secret Sharing)

Message space: finite field $\mathbb{F}$

let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}^n$ be some fixed constants

→ Share(m): pick a random degree-t polynomial, s.t.,
$S(0) = m$



$$\Rightarrow S(x) = m + \sum_{i=1}^{t} c_i x^i$$

$S_1 = S(\alpha_1)$, $S_2 = S(\alpha_2)$, $\ldots$, $S_n = S(\alpha_n)$

→ Reconstruct $(S_1, \ldots, S_{t+1})$: Lagrange interpolation to find $S(0) = m$.

Semi-Honest Secure Multi-Party Computation

Definition: A protocol π securely computes a function f in the semi-honest model, if $\exists$ a PPT simulator algorithm S, s.t., $\forall$ t-sized subset $C \subset [n]$ of corrupt parties, for any security parameter λ & $\forall$ inputs $x_1, \ldots, x_n$, it holds that:

$$\left\{ S\left(\{x_i\}_{i \in C}, f(x_1, \ldots, x_n) \right), f(x_1, \ldots, x_n) \right\} \approx_c \left\{ \underbrace{\mathrm{View}_C(\pi)}_{\text{View of Adv}}, \underbrace{\mathrm{Out}_{[n] \setminus C}(\pi)}_{\text{output of honest parties}} \right\}$$

Semi-Honest MPC: BGW Protocol (1988)

Michael Ben-Or

Shafi Goldwasser

Avi Wigderson

- At most $t < n/2$ semi-honest corruptions
- Information-theoretically secure.

BGW Protocol

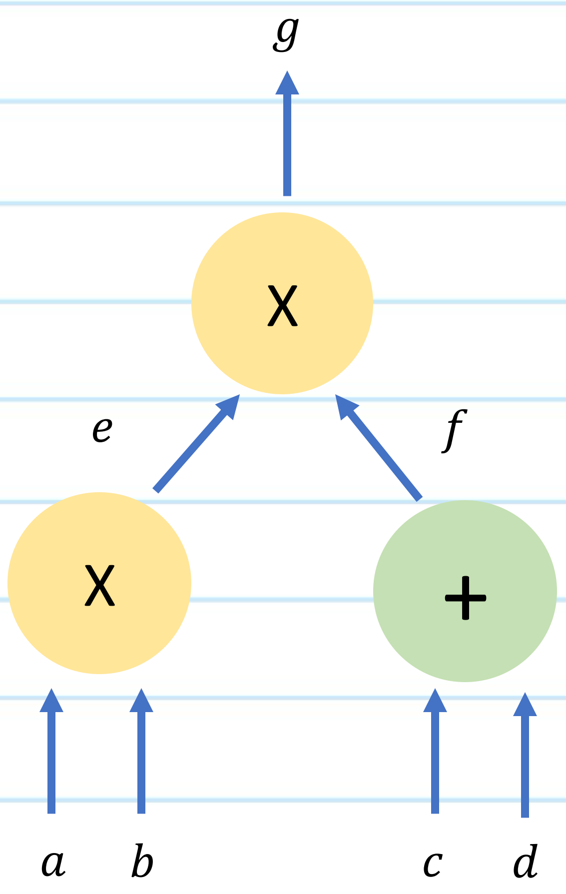
→ Let the function that the parties wish to compute be $f: \mathbb{F}^n \rightarrow \mathbb{F}^k$

→ We assume that all parties have an arithmetic circuit representing the function f.

→ Similar to GMW, this protocol proceeds in three phases:

1) Input Sharing
2) Circuit Evaluation
3) Output Reconstruction.

BGW Protocol : Overview



* Input Sharing: Parties start by computing and sending (t,n) threshold shares of their respective inputs.

* Circuit Evaluation: gate-by-gate evaluation over secret-shared values. In other words, compute secret-share of all intermediate wire values one by one

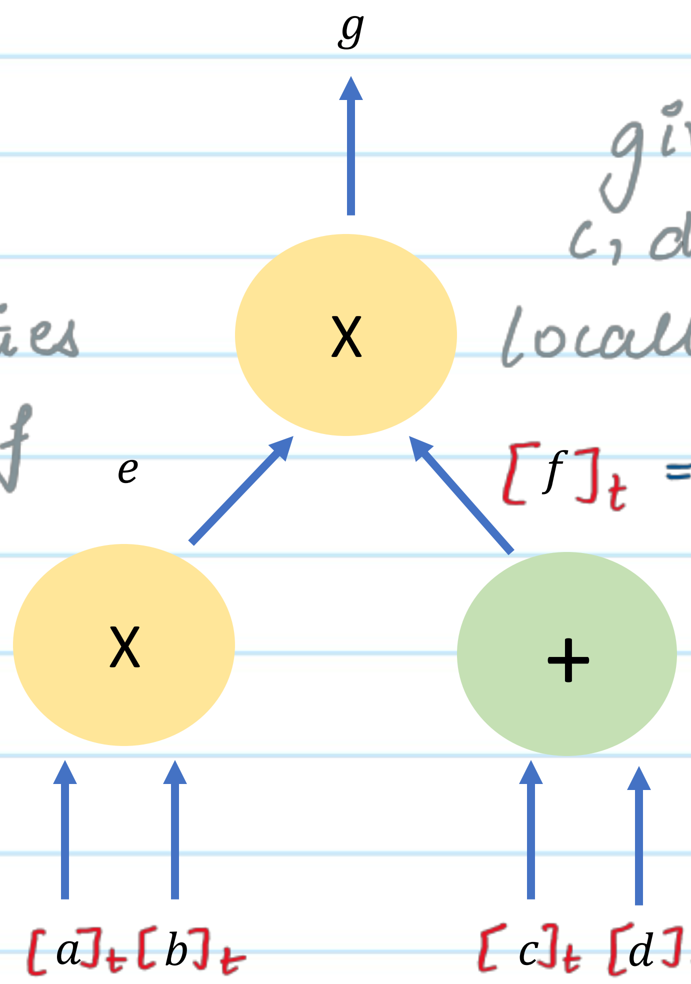
* Output Reconstruction: All parties reveal their shares of the output wire values to each other & then reconstruct.

BGW Protocol

Multiplication

given shares of a, b, the parties need to compute shares of
$e = a \times b$

Can the parties simply locally multiply their respective shares of a & b?



Input sharing using a (t,n) Shamir-sharing

Addition

given shares of c, d, parties can locally compute:

$[f]_t = [c]_t + [d]_t$

BGW Protocol: Multiplication Gates.

Given: $[a]_t$, $[b]_t$ **To compute:** $[e = a\cdot b]_t$

$\forall i \in [n]$, Each party P_i does the following:

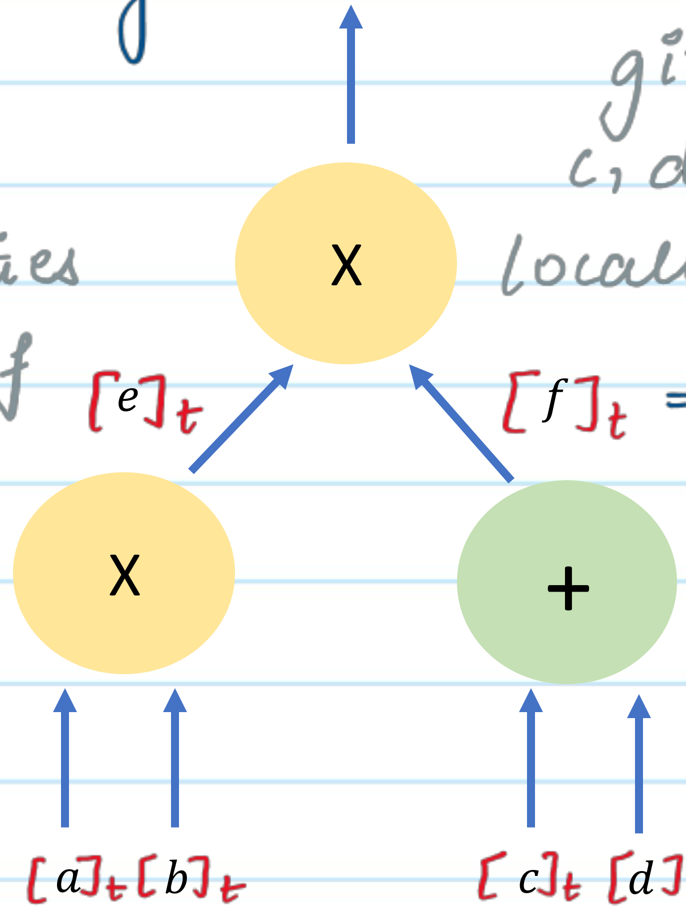
1. locally computes $\bar{e}_i = a_i \times b_i$
2. Computes (t,n) Shamir Sharing of $\bar{e}_i$
   $(\bar{e}_{i1}, \ldots, \bar{e}_{in}) \leftarrow \text{Share}(\bar{e}_i)$
3. $\forall j \in [n]$, Send $\bar{e}_{ij}$ to Party P_j
4. Let $\lambda_1, \ldots \lambda_n$ be Lagrange coefficients
   such that $ab = \lambda_1 \bar{e}_1 + \lambda_2 \bar{e}_2 + \cdots \lambda_n \bar{e}_n$
   Party P_i computes $ab_i = \lambda_1 \bar{e}_{1i} + \lambda_2 \bar{e}_{2i} + \cdots + \lambda_n \bar{e}_{ni}$

1. $[e]_{2t} = [a]_t \times [b]_t$
2. $[e]_{2t} \rightarrow [[e]_{2t}]_t$
3. exchange shares of shares
4. $[[e]_{2t}]_t \rightarrow [e]_t$

BGW Protocol

Reconstruct Output.

$g \leftarrow [g]_t$



Addition

given shares of c, d, parties can locally compute:

$[f]_t = [c]_t + [d]_t$

Multiplication

given shares of a, b, the parties need to compute shares of

$e = a \times b$

$[e]_{2t} = [a]_t \times [b]_t$

$[[e]_{2t}]_t \xleftarrow{\text{share}} [e]_{2t}$

exchange $[[e]_{2t}]_t$

$[e]_t \xleftarrow{\text{reconstruct}} [[e]_{2t}]_t$

Input sharing using a (t,n) Shamir-sharing

BGW Protocol : Security

What do we want to Prove?

→ BGW is an n-party protocol for securely computing f in the presence of a semi-honest adversary who corrupts at most $t < n/2$ parties.

→ $\exists$ a simulator, s.t. for any t-sized subset $C \subset [n]$ of corrupt parties and $\forall\, x_1, \ldots, x_n$, it can simulate a view using inputs of the corrupt parties & output of f that is indistinguishable from the adversary's view in the real protocol.

→ for simplicity, let's consider an adv who corrupts exactly $n/2 - 1$ parties.

Simulator: $S_c\left(\{x_i\}_{i \in C},\ f(x_1, \ldots, x_n)\right\}$

1 Input Sharing: $\forall i \in C$, compute $\text{Share}(x_i)$
$\forall i \notin C$, compute $\text{Share}(0)$

2. Circuit Evaluation: $\forall i \in C$:

→ Addition $(f = c + d)$: compute $f_i = c_i + d_i$

→ Multiplication $(e = a \times b)$: compute $\bar{e}_i = a_i \times b_i$

compute $\text{Share}(\bar{e}_i)$

$\forall j \notin C$, sample $\bar{e}_{ji} \xleftarrow{\$} \mathbb{F}$

compute

$ab_i = \lambda_1 \bar{e}_{1i} + \ldots + \lambda_n \bar{e}_{ni}$

Simulator: $S_C(\{x_i\}_{i\in C}, f(x_1,\dots,x_n))$

3. Output Reconstruction: For each output wire y, let $\{y_i\}_{i\in C}$ be the shares that the simulator computed during circuit eval.

Interpolate $(y, \{y_i\}_C)$ to reconstruct a polynomial $p(x)$ such that $p(0)=y$.

$\forall j \in [n]\setminus C: \quad y_j = y(\alpha_j)$.

Exercise: Why is the view generated by this simulator *perfectly* indistinguishable from adversary's view.