CS 65500
Advanced Cryptography

Lecture 10: Shamir Secret Sharing and MPC

Instructor: Aarushi Goel

Spring 2025

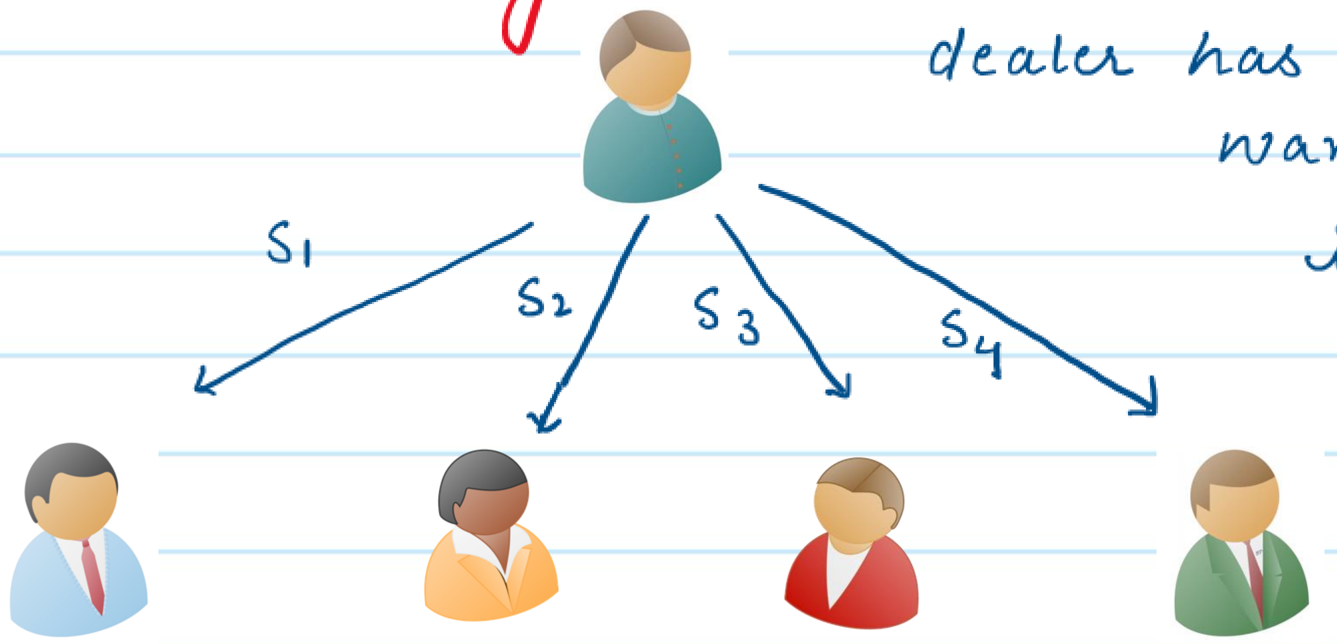
Agenda

→ Threshold Shamir Secret Sharing

→ Secure Multiparty Computation

Reminder: HW3 will be released today!

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)

Definition: A (t,n) secret sharing consists of a pair of PPT algorithms (Share, Reconstruct) s.t.,

- $\text{Share}(s) \rightarrow (s_1, \ldots, s_n)$

- $\text{Reconstruct}(s'_{i_1}, \ldots, s'_{i_{(t+1)}})$ is such that, if $\{s'_{i_1}, \ldots, s'_{i_{(t+1)}}\} \subseteq \{s_1, \ldots, s_n\}$, then it outputs s.

- $\forall s, s'$ and for any subset of at most t indices $X \subset [1, n]$, $|X| \leq t$ the following distributions are statistically close:

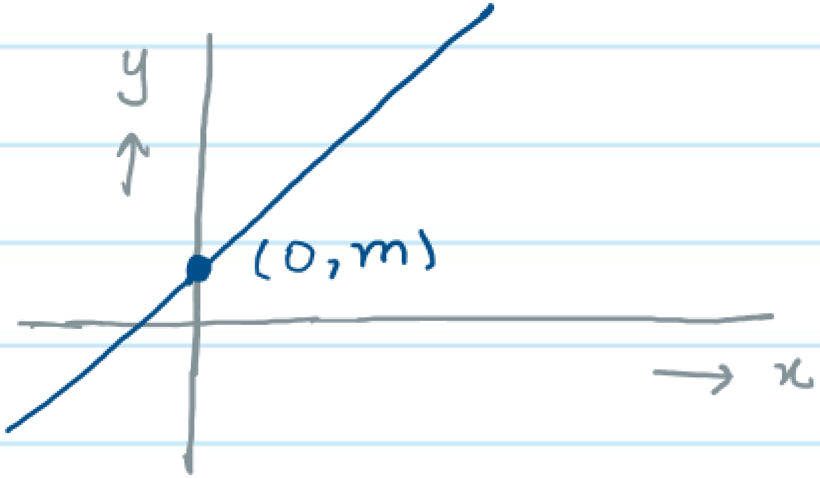
$$\{(s_i \mid i \in X) \;;\; (s_1, \ldots, s_n) \leftarrow \text{Share}(s)\},$$

$$\{(s'_i \mid i \in X) \;;\; (s'_1, \ldots, s_n) \leftarrow \text{Share}(s')\}$$

Construction: (1, n) Threshold 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

$r \xleftarrow{\$} \mathbb{F}$

$$s(x) = rx + m$$

$$s_1 = s(\alpha_1),\ s_2 = s(\alpha_2),\ \ldots,\ s_n = s(\alpha_n)$$

→ **Reconstruct(s_i, s_j):** $r = \dfrac{(s_i - s_j)}{(\alpha_i - \alpha_j)}$ $\quad m = s_i = r \cdot \alpha_i$

Construction: $(1, n)$ Threshold Secret Sharing

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

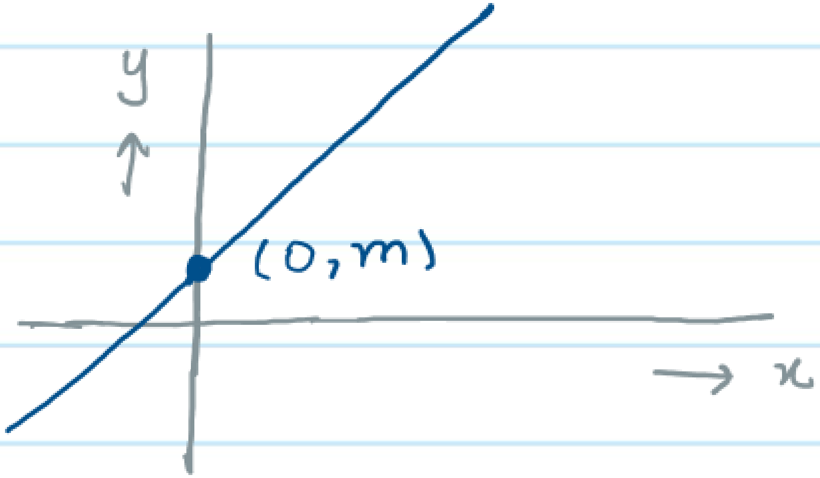
Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}^n$ be some fixed constants

→ Share (m): pick a random

$r \xleftarrow{\$} \mathbb{F}$

$s(x) = rx + m$

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



Is each S_i by itself uniformly distributed, irrespective of m? why?

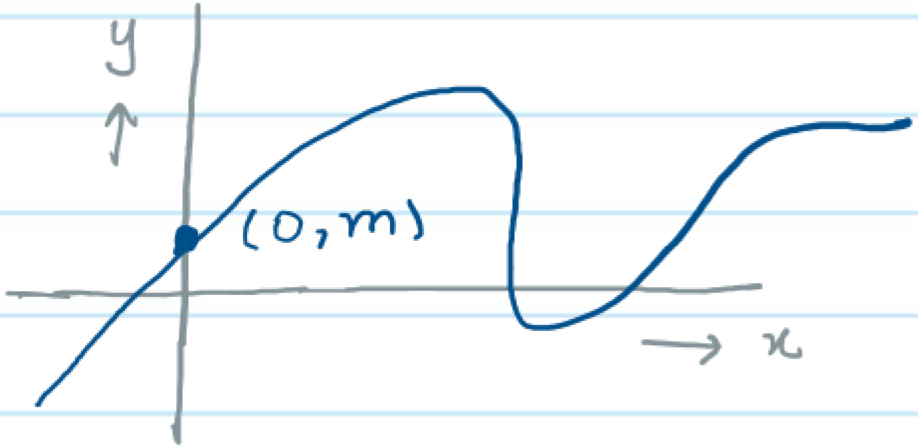
Construction: (t, n) Threshold Secret Sharing
(Shamir Secret Sharing)

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 $S(0) = m$.

Linear Secret Sharing Scheme

→ Share (m) :

$$\underbrace{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}_{n \times (t+1)} \underbrace{\begin{bmatrix} m \\ \left.\begin{matrix} c_1 \\ \vdots \\ c_t \end{matrix}\right\} \text{randomness} \end{bmatrix}}_{(t+1) \times 1} = \underbrace{\begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}}_{n \times 1}$$

→ Reconstruction : $\forall\ X \subset [n]$, $|X| = t+1$, $\exists$ a vector , s.t.

$$\begin{bmatrix} & & & \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_{t+1} \end{bmatrix} = m$$

Shamir is a Linear Secret Sharing Scheme

→ Share (m) :

$$\underbrace{\begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^t \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha_n & \cdots & \cdots & \alpha_n^t \end{bmatrix}}_{n \times (t+1)} \underbrace{\begin{bmatrix} m \\ c_1 \\ \vdots \\ c_t \end{bmatrix}}_{(t+1) \times 1} = \underbrace{\begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}}_{n \times 1}$$

Vandermonde Matrix

→ Reconstruction: $\forall\ X \subset [n]$, $|X| = t+1$, $\exists$ a vector, s.t.

$$\boxed{} \begin{bmatrix} s_1 \\ \vdots \\ s_{t+1} \end{bmatrix} = m$$

polynomial interpolation

Computing on Linear Shares

Suppose two secrets m_1 and m_2 were shared using the same secret-sharing scheme

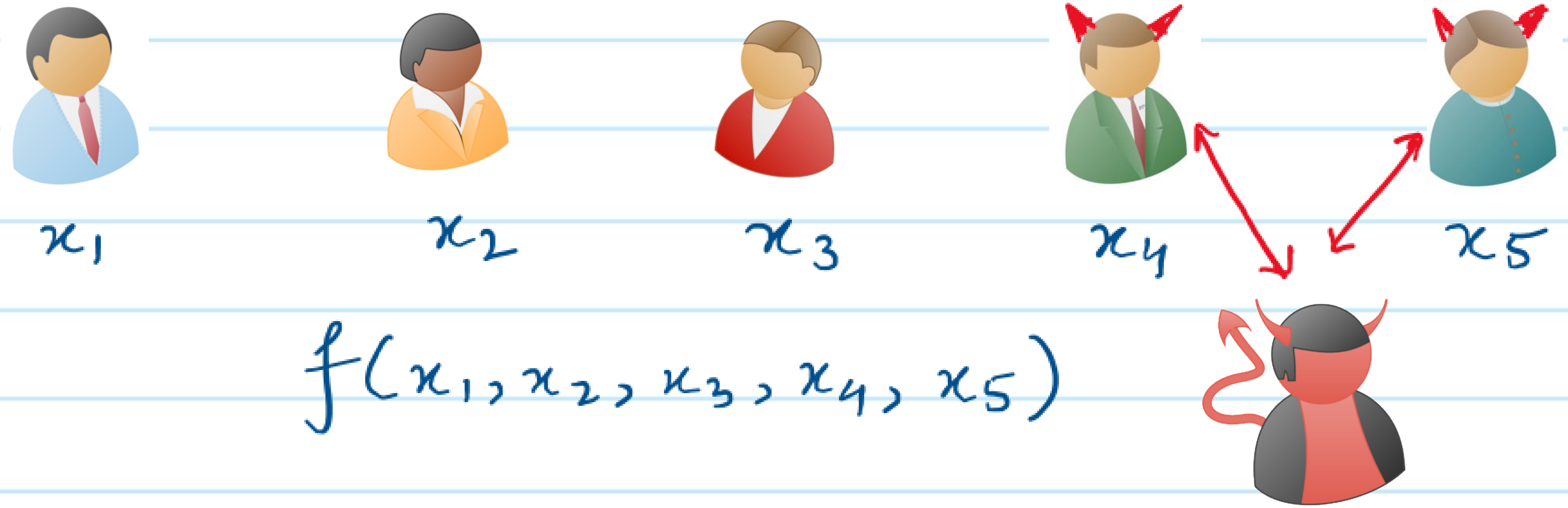
$$\begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix} \begin{bmatrix} m_1 \\ c_1 \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \qquad \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix} \begin{bmatrix} m_2 \\ d_1 \\ \vdots \\ d_t \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}$$

Then for any $p, q \in \mathbb{F}$, shares of $p \cdot m_1 + q \cdot m_2$ can be computed locally by each party i as:

$$p \cdot s_i + q \cdot r_i$$

Secure Multi-Party Computation : Setting

Parties: P_1 P_2 P_3 P_4 P_5



Inputs: x_1 x_2 x_3 x_4 x_5

$$f(x_1, x_2, x_3, x_4, x_5)$$

→ Suppose t out of n parties are corrupt.

→ We will assume that all corrupt parties are controlled by a monolithic *adversary*

→ Can the parties securely compute f on their joint inputs?

Secure Multi-Party Computation : Security

→ An adversary corrupting at most t out of the n parties learn nothing about the inputs of the remaining *honest* parties beyond what is already revealed by the output of the function.

→ In other words, whatever the adversary sees in the protocol, it could have simulated himself using only the inputs of the corrupt parties and output of the function.

Formalizing the Security Requirements for Secure MultiParty Computation

{View of the adversary in the protocol}

is indistinguishable from

{A simulated view that the adversary could have computed himself given inputs of the corrupted parties and output of the function, without having communicated with the honest parties}

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:

$$\left\{ S\left(\{x_i\}_{i \in C}, f(x_1, \ldots, x_n) \right), f(x_1, \ldots, x_n) \right\} \approx_c$$

$$\left\{ \underbrace{\mathrm{View}_C(\pi)}_{\text{View of Adv}}, \underbrace{\mathrm{Out}_{[n] \setminus C}(\pi)}_{\text{output of honest parties}} \right\}$$