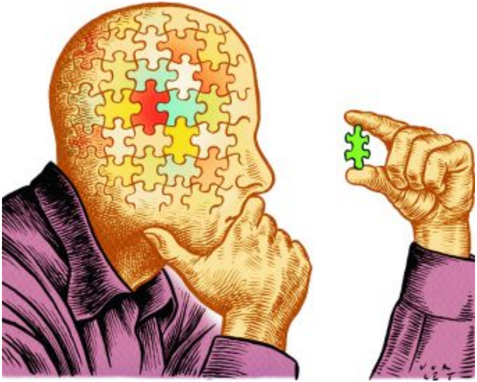
Secret Sharing & SMPC - RECAP



Disclaimer

This is a *very quick* recap, very result-oriented! I ‘fly over’ many important aspects of Secret Sharing and Secure Muliparty Computation.

Recap aim: show how to use MPC to compute the sum of two secret values

Roadmap:

1. Recall what MPC is about.
2. Recall what Secret Sharing is about.
3. Example of how Shamir Secret Sharing Scheme works.
4. Example of how to use Shamir SSS to do MPC.

Disclaimer 2

The example is ‘small’ and ‘quick’ but not secure (can you tell why?). Its only purpose it to show how things work in a simple context.

Multiparty Computation

In a Multiparty Computation Protocol there are:

- n participants $\mathrm{P}_1, \cdots, \mathrm{P}_\mathrm{n}$,
- n inputs x_i (one for each participant P_i),
- a function f that the participants want to evaluate on all the inputs (i.e. the goal is to compute $y = f(x_1, \cdots, x_n)$).

Essential Properties

1. **Correctness:** the correct value of y is computed; and
2. **Privacy:** y is the only new information that is released (i.e. P_i shall not learn the input x_j for of participant P_j if $j \neq i$)

Attacks goals

The attacker aim is either to *learn private information* (e.g. the inputs x_i) or to *cause incorrect computations* (output a value $y^* \neq y = f(x_1, \ldots, x_n)$).

Question: How to keep input private in MPC? Use **Secret Sharing Methods**.

Secret Sharing Schemes

A **secret-sharing scheme** usually involves:

- a *dealer* D who has a *secret* s,
- n *parties* $\mathrm{P}_1, \ldots, \mathrm{P}_\mathrm{n}$.

A secret-sharing scheme is a method by which the dealer distributes shares of s to the n parties such a way that:

(1) any subset of $k+1$ parties can reconstruct the secret from its shares

and

(2) any subset of k parties cannot retrieve any partial information on the secret s.

Shamir's Secret Sharing Scheme

Shamir's Secret Sharing Scheme

In order to share a secret $s \in \mathbb{Z}_p$ among n parties in such a way that any subset of $k+1$ party can recover s, but no subset of k succeeds in retreiving the secret, a Dealer D performs the following steps:

1. select k random values $a_i \xleftarrow{\$} \mathbb{Z}_p$ and construct the polynomial

$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_k x^k \in \mathbb{Z}_p[x]$$

2. evaluate f on the points $i = 1, 2, \ldots, n$, and send to the i-th participant $\mathrm{P_i}$ its share $f_i = f(i) \in \mathbb{Z}_p$

Note 1: the secret $s = f(0)$ is the constant term of f.
Note 2: the above procedure holds for any field (not only $\mathbb{Z}_p$), in particular you can follow this recipe also on $\mathbb{R}$.

Shamir's Secret Sharing Scheme

Question: How to recover the secret s from the f_is?

Lagrange Interpolation

For any polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree k it holds that

$$f(x) = f(1)\delta_1(x) + f(2)\delta_2(x) + \ldots + f(k+1)\delta_{k+1}(x) \in \mathbb{Z}_p[x] \ ,$$

where the $\delta_i(x) \in \mathbb{Z}_p[x]$ are the degree-k interpolation polynomials defined as:

$$\delta_i(x) = \prod_{j=1, j\neq i}^{k+1} \frac{x-j}{i-j} \ .$$

Note: the $\delta_i(x)$ only depend on $i, j \in \{1, 2, \ldots, n\}$ and not on the polynomial f.

Therefore any set of $k+1$ parties can compute $s = f(0)$ if all parties jointly compute the Lagrange interpolation.

Secure Multiparty Computation - Example

Secure Multiparty Computation - Example

Setting: two dealers, D_1 and D_2, have each one share (s_1 and s_2 respectively) and want to securely compute $s_1 + s_2$.

Recipe:

1. D_i uses Shamir's SSS to share s_i among n parties
2. each party P_i holds two shares, f_i from D_1 and g_i from D_2.
3. each party P_i locally computes $s(i) = f_i + g_i = f(i) + g(i) = (f + g)(i)$.
4. any subset of $k + 1$ parties now can jointly compute $s_1 + s_2$: by using Lagrange interpolation between the $s(i)$. I.e. the parties compute $h(x) = (f + g)(x)$ from the values $s(i)$ and the interpolation polynomials $\delta_i(x)$, the final result is $h(0) = s_1 + s_2$.

Numerical Example

D$_1$'s secret is $s_1 = 3$

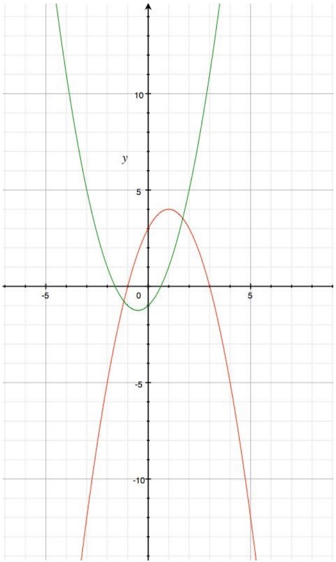
polynomial for secret sharing:
$f(x) = 3 + 2x - x^2$

shares (among 3 participants):
$f_1 = f(1) = 4, f_2 = f(2) = 3, f(3) = 0$

D$_2$'s secret is $s_2 = -1$

polynomial for secret sharing:
$g(x) = -1 + x + x^2$

shares (among 3 participants):
$g_1 = g(1) = 1, g_2 = g(2) = 5, g_3 = g(3) = 11$



Numerical Example

Local addition of the shares:

$P_1 : h(1) = f_1 + g_1 = 5$,
$P_2 : h(2) = f_2 + g_2 = 8$,
$P_1 : h(3) = f_3 + g_3 = 11$.

Lagrange interpolation:

$\delta_1(x) = \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} = \frac{x^2-5x+6}{2}$,
$\delta_2(x) = \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} = -(x^2 - 4x + 3)$,
$\delta_3(x) = \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} = \frac{x^2-3x+2}{2}$.

Thus: $h(x) = \sum_{i=1}^{3} h(i)\delta_i(x) = 3x + 2$
And $h(0) = s_1 + s_2 = 2$.