Secret Sharing

Patrick Ah-Fat

Imperial College London

pwa14@ic.ac.uk

Overview

1. Secret Sharing
   - Introduction
   - Monotone Access Structures
   - Adversarial Models
2. Threshold Scheme
   - Polynomial Interpolation
   - Shamir Secret Sharing Scheme
   - Lagrange Interpolation

Objective of **Secret Sharing**:

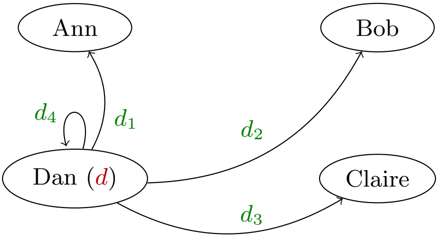
- **Store** sensitive data d

Problems:

- Data can be **stolen** (loss and leakage)
- Data can be **corrupted** or **damaged**

Idea:

- **Split** data d into shares $d_1, \cdots, d_4$
- Store shares into **different locations** (send to different parties)
- Secret **cannot be recovered** if one or few shares are stolen
- Secret **can be recovered** if some shares are missing or corrupted



Secret Sharing scheme characterised by its **access structure** Γ: all subsets of parties in $\mathcal{P}$ that are able to recover the secret (such subsets are called **qualified**).

Example: $\mathcal{P} = \{Ann, Bob, Claire, Dan\}$.
All sets containing $\{Ann, Bob\}$ or $\{Ann, Claire, Dan\}$ are qualified. Then:

$$\Gamma = \{\{A, B\}, \{A, B, C\}, \{A, B, D\}, \{A, B, C, D\}, \{A, C, D\}\}$$

We can characterise Γ by the set of its **minimal elements** $m(\Gamma)$ (with respect to subset inclusion):

$$m(\Gamma) = \{\{A, B\}, \{A, C, D\}\}$$

We can assume Γ to be **monotone** with respect to subset inclusion: if $\{Ann, Bob\}$ can recover the secret, then $\{Ann, Bob, Claire\}$ can too.

Let $\Gamma \subseteq \mathscr{P}(\mathcal{P})$ be a collection of subsets of a finite set $\mathcal{P}$. Then Γ is a monotone access structure iff:

- Γ is non-empty (some parties can recover the secret).
- $\forall A \subseteq B \subseteq \mathcal{P}\colon (A \in \Gamma) \implies (B \in \Gamma)$ (closure under supersets).

We note that necessarily $\mathcal{P} \in \Gamma$.

Set of minimal elements of Γ w.r.t. subset inclusion is denoted by $m(\Gamma)$.

Schemes for **general** access structures:

- Ito-Saito-Nishizeki
- Replicated Secret Sharing scheme

Problems:

- inefficient in terms of the **number of shares** needed to distribute a secret
- do not naturally adapt to the presence of **dishonest parties**

Let $1 \leq t \leq n$ be integers and $\mathcal{P} = \{P_1, \cdots, P_n\}$ be a set of n parties. The t-out-of-n monotone access structure Γ is defined as:

$$m(\Gamma) = \{S \subseteq P \mid |S| = t\}$$

or equivalently:

$$\Gamma = \{S \subseteq P \mid |S| \geq t\}$$

Efficient scheme and resilient against dishonest parties: Shamir secret sharing scheme.

Passive adversary (honest-but-curious): Abides by the protocol, but shares with other adversaries all the information that he receives during the protocol so as to **infer as much information as possible** on the secrets.

Active adversary (malicious): Can also **deviate from the protocol** by sending erroneous data and cooperating with other adversaries (in order to corrupt the protocol output or learn more information on the secrets).

Example: Ann splits her secret $a = a_1 + a_2$ and sends a_1 to Bob, a_2 to Claire.

- Passive attacker Bob alone cannot infer anything about a.
- Passive attackers Bob and Claire can recover secret a.
- Active attacker Bob can corrupt secret a by corrupting his share a_1.

Let $0 \leq t < n$ be two integers and $\mathcal{P}$ be a set of n parties.

Shamir secret sharing scheme (threshold scheme)

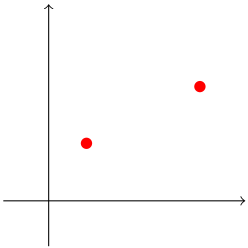
- Share a secret s amongst the n parties.
- $t+1$-out-of-n scheme: any $t+1$ parties can together recover secret s
- **But** any t parties together cannot learn any information on s (the scheme thus allows up to t passive adversaries).
- Efficient in terms of required number of shares (1 per party).

How ?

- Polynomial interpolation (Lagrange)

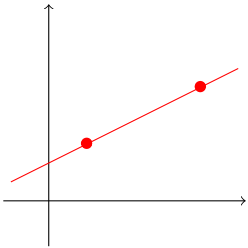
Theorem

By t points passes one and only one polynomial of degree at most $t-1$.



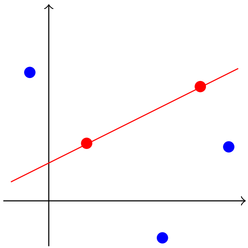
Theorem

By t points passes one and only one polynomial of degree at most $t-1$.



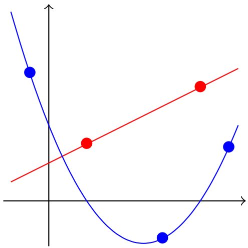
Theorem

By t points passes one and only one polynomial of degree at most $t - 1$.



Theorem

By t points passes one and only one polynomial of degree at most $t - 1$.

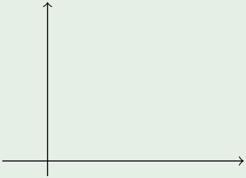


Note: Such a polynomial can be efficiently recovered.

Question: How to build a threshold secret sharing scheme out of this ?

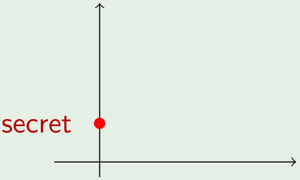
Example

Aim: Share secret 1 amongst $P_1, \cdots, P_4$ with 2 out of 4 threshold.



Example

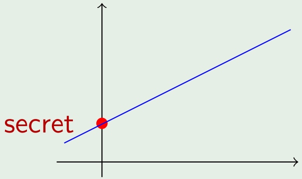
Aim: Share secret 1 amongst $P_1, \cdots, P_4$ with 2 out of 4 threshold.



- Place $(0, 1)$.

Example

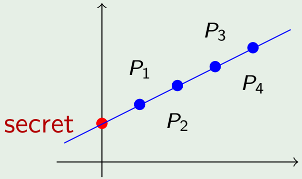
Aim: Share secret 1 amongst $P_1, \cdots, P_4$ with 2 out of 4 threshold.



- Place $(0, 1)$.
- Choose a random polynomial f of degree ≤ 1 passing through $(0, 1)$.

Example

Aim: Share secret 1 amongst $P_1, \cdots, P_4$ with 2 out of 4 threshold.



- Place $(0, 1)$.
- Choose a random polynomial f of degree ≤ 1 passing through $(0, 1)$.
- Send each $f(i)$ to party P_i.

Shamir Secret Sharing Scheme

Aim: Share secret s amongst n parties with up to t adversaries.

- Choose random polynomial f of degree $\leq t$ such that $f(0) = s$.
- Send $f(i)$ to each P_i.

We say that the parties share secret s via a polynomial f of degree at most t and we write: $[s, f]_t$.

Shamir Secret Sharing Scheme

Aim: Share secret s amongst n parties with up to t adversaries.

- Choose random polynomial f of degree $\leq t$ such that $f(0) = s$.
- Send $f(i)$ to each P_i.

We say that the parties share secret s via a polynomial f of degree at most t and we write: $[s, f]_t$.

Note: We place ourselves in $\mathbb{Z}_p$.
Claim 1: Any set of $\leq t$ parties cannot infer anything on s.
Claim 2: Any set of $> t$ parties can recover s.

Shamir Secret Sharing Scheme

Aim: Share secret s amongst n parties with up to t adversaries.

- Choose random polynomial f of degree $\leq t$ such that $f(0) = s$.
- Send $f(i)$ to each P_i.

We say that the parties share secret s via a polynomial f of degree at most t and we write: $[s, f]_t$.

Note: We place ourselves in $\mathbb{Z}_p$.
Claim 1: Any set of $\leq t$ parties cannot infer anything on s.
Claim 2: Any set of $> t$ parties can recover s. **How ?**

Setting:

- Finite field $\mathbb{F} = \mathbb{Z}_p$
- $t \in \mathbb{N}$ (maximal number of adversaries)
- $Z \subseteq \mathbb{Z}_p$ such that $|Z| > t$ (parties willing to recover the secret)
- $P \in \mathbb{Z}_p[X]$ such that $\deg(P) \leq t$

Question: Given $\Big(P(i)\Big)_{i \in Z}$, how to recover P, and in particular $P(0)$?

Definition (Lagrange Polynomial)

$$\delta_i(X) = \prod_{\substack{j \in Z \\ j \neq i}} \frac{X - j}{i - j} \qquad Q(X) = \sum_{i \in Z} \delta_i(X) \cdot P(i)$$

Note:

- $\begin{cases} \forall i \in Z : \delta_i(i) & = & 1 \\ \forall i \neq k \in Z : \delta_i(k) & = & 0 \end{cases}$

Definition (Lagrange Polynomial)

$$\delta_i(X) = \prod_{\substack{j \in Z \\ j \neq i}} \frac{X - j}{i - j} \qquad\qquad Q(X) = \sum_{i \in Z} \delta_i(X) \cdot P(i)$$

Note:

- $\begin{cases} \forall i \in Z \colon \delta_i(i) & = & 1 \\ \forall i \neq k \in Z \colon \delta_i(k) & = & 0 \end{cases}$
- $\deg(P), \deg(Q) < |Z|$ and P and Q agree on $|Z|$ points (of Z)

Definition (Lagrange Polynomial)

$$\delta_i(X) = \prod_{\substack{j \in Z \\ j \neq i}} \frac{X - j}{i - j} \qquad Q(X) = \sum_{i \in Z} \delta_i(X) \cdot P(i)$$

Note:

- $\begin{cases} \forall i \in Z \colon \delta_i(i) & = & 1 \\ \forall i \neq k \in Z \colon \delta_i(k) & = & 0 \end{cases}$
- $\deg(P), \deg(Q) < |Z|$ and P and Q agree on $|Z|$ points (of Z)
- Fundamental Theorem of Algebra $\implies P = Q$

Conclusion: $s = P(0) = Q(0) = \sum_{i \in Z} \delta_i(0) \cdot P(i)$

Definition (Lagrange Polynomial)

$$\delta_i(X) = \prod_{\substack{j \in Z \\ j \neq i}} \frac{X - j}{i - j} \qquad\qquad Q(X) = \sum_{i \in Z} \delta_i(X) \cdot P(i)$$

Note:

- $\begin{cases} \forall i \in Z\colon \delta_i(i) & = & 1 \\ \forall i \neq k \in Z\colon \delta_i(k) & = & 0 \end{cases}$
- $\deg(P), \deg(Q) < |Z|$ and P and Q agree on $|Z|$ points (of Z)
- Fundamental Theorem of Algebra $\implies P = Q$

Conclusion: $s = P(0) = Q(0) = \sum_{i \in Z} \delta_i(0) \cdot P(i)$

Definition (Recombination Vector)

Let $r = \left(\delta_i(0)\right)_{i \in Z}$. If $\deg(P) < |Z|$, then: $s = \sum_{i \in Z} r_i \cdot P(i)$.

Example

Sharing: We place ourselves in $\mathbb{F} = \mathbb{Z}_{11}$.
Let's share $s = 5$ amongst $n = 5$ parties with up to $t = 2$ adversaries.
We choose random $f(X) = 5 + 3X + 8X^2$.
We compute $\Big(f(i)\Big)_{1 \le i \le 5} = (5, 10, 9, 2, 0)$, and send $f(i)$ to P_i.

Example

Sharing: We place ourselves in $\mathbb{F} = \mathbb{Z}_{11}$.
Let's share $s = 5$ amongst $n = 5$ parties with up to $t = 2$ adversaries.
We choose random $f(X) = 5 + 3X + 8X^2$.
We compute $\Big(f(i)\Big)_{1 \leq i \leq 5} = (5, 10, 9, 2, 0)$, and send $f(i)$ to P_i.

Recovering: Let's assume $Z = \{2, 3, 5\}$. Let's compute $r = \Big(\delta_i(0)\Big)_{i \in Z}$.
We have:

$$r_2 = \delta_2(0) = \prod_{\substack{j \in Z \\ j \neq 2}} \frac{-j}{2-j} = \frac{-3}{2-3} \cdot \frac{-5}{2-5} = 5$$

So $r_2 = 5$.

Example

Sharing: We place ourselves in $\mathbb{F} = \mathbb{Z}_{11}$.
Let's share $s = 5$ amongst $n = 5$ parties with up to $t = 2$ adversaries.
We choose random $f(X) = 5 + 3X + 8X^2$.
We compute $\Big(f(i)\Big)_{1 \le i \le 5} = (5, 10, 9, 2, 0)$, and send $f(i)$ to P_i.

Recovering: Let's assume $Z = \{2, 3, 5\}$. Let's compute $r = \Big(\delta_i(0)\Big)_{i \in Z}$.
We have:

$$r_3 = \delta_3(0) = \prod_{\substack{j \in Z \\ j \neq 3}} \frac{-j}{3 - j} = \frac{-2}{3 - 2} \cdot \frac{-5}{3 - 5} = 6$$

So $r_2 = 5$, $r_3 = 6$.

Example

Sharing: We place ourselves in $\mathbb{F} = \mathbb{Z}_{11}$.
Let's share $s = 5$ amongst $n = 5$ parties with up to $t = 2$ adversaries.
We choose random $f(X) = 5 + 3X + 8X^2$.
We compute $\Big(f(i)\Big)_{1 \le i \le 5} = (5, 10, 9, 2, 0)$, and send $f(i)$ to P_i.

Recovering: Let's assume $Z = \{2, 3, 5\}$. Let's compute $r = \Big(\delta_i(0)\Big)_{i \in Z}$.
We have:

$$r_5 = \delta_5(0) = \prod_{\substack{j \in Z \\ j \neq 5}} \frac{-j}{5-j} = \frac{-2}{5-2} \cdot \frac{-3}{5-3} = 1$$

So $r_2 = 5,\ r_3 = 6,\ r_5 = 1$.

Example

Sharing: We place ourselves in $\mathbb{F} = \mathbb{Z}_{11}$.
Let's share $s = 5$ amongst $n = 5$ parties with up to $t = 2$ adversaries.
We choose random $f(X) = 5 + 3X + 8X^2$.
We compute $\Big(f(i)\Big)_{1 \le i \le 5} = (5, 10, 9, 2, 0)$, and send $f(i)$ to P_i.

Recovering: Let's assume $Z = \{2, 3, 5\}$. Let's compute $r = \Big(\delta_i(0)\Big)_{i \in Z}$.

So $r_2 = 5,\ r_3 = 6,\ r_5 = 1$. And thus:

$$s = \sum_{i \in Z} r_i \cdot f(i) = 5 * 10 + 6 * 9 + 1 * 0 = 5$$

- Secret Sharing: keeping a secret **safe** and **secure**
- Split a secret into several **shares** sent to different parties
- Monotone **access structures** characterise SS schemes
- **Threshold** scheme: Shamir Secret Sharing Scheme
- Safety and security guaranteed by **Lagrange interpolation**