lec 09

Generalized Secret sharing (GSS) 1987

If an access structure can be demonestrated by a monotone formula (i.e., a formula with "or" and "and" gates without any "not" gates), it can be formalized by an efficient generalized secret sharing scheme.

Seret sharing

1. Initially, the access structure is defined by the dealer as a montone formula F. Let $S(\alpha, F)$ be a random function for secret "α" & a monotone formula "F", which is defined as follows:

$$S(\alpha, \vee(F_1, F_2, \ldots, F_n)) = \bigcup_{1 \leqslant i \leqslant n} S(\alpha, F_i)$$

$$S(\alpha, \wedge(F_1, F_2, \ldots, F_n)) = \bigcup_{1 \leqslant i \leqslant n} S(\alpha_i, F_i), \text{ where } \alpha = \sum_{i=1}^{n} \alpha_i \pmod p$$

and α_i are chosen uniformly at random from the FF $\mathbb{Z}_p$.

2. The dealer moves over "or" gates in order to define different sets of players who can recover the secret independently based on the defined access structure. He shares α (secret) among the players of each set, who are connected by "and" gates such that the summation of the shares is equal to $\alpha \pmod p$ in each set.

Secret Recovery

1. the players in each set can then add their shares together to reconstruct the secret "α" independent of other sets of players.

$\boxed{\text{Example:}}$ If it is desirable to divide a secret among 4 players P_i in such a way that either P_1 together with P_2 or P_3 together with P_4 can reconstruct the secret, the monotone formula would be $((P_1 \wedge P_2) \vee (P_3 \wedge P_4))$.

To share 'α' according to this access structure, the dealer shares the secret independently between these two sets $\longrightarrow$

$$\alpha = \alpha_1 + \alpha_2 \pmod{p}$$

$$\alpha = \alpha_3 + \alpha_4 \pmod{p}$$

$\longrightarrow$ α_i are random numbers from $\mathbb{Z}_p$

Conditions: ⓐ To recover the secret, the total weight of authorized players $\in \Delta$ (uncorrupted) must be equal or greater than the threshold:

$$\sum_{P_i \in \Delta} w_i \geqslant t$$

ⓑ On the other hand, the total weight of Colluders $\in \nabla$ (corrupted) must be less than the threshold:

$$\sum_{P_i \in \nabla} w_i < t$$

ⓒ Finally, the weight of each player is bounded to a parameter much less than t:

$$w_i \leqslant m \ll t \quad \text{for } 1 \leqslant i \leqslant n$$

$m = 4 \longrightarrow$ maximum weight of each player

P_1	1	2	3	4
P_2	5	.	.	8
⋮				
P_n	.	.	.	.

secret sharing

① Similar to TSS: $\begin{cases} f(x) \in \mathbb{Z}_p[x] & \text{of degree } t-1 \\ f(0) = \alpha & \text{secret} \end{cases}$

② shares of player P_i for $1 \leq i \leq n$ will be defined based on his weight w_i

$$s_{ij} = f(\nu_{ij}) \quad \text{for} \quad 1 \leq j \leq w_i$$

where $\nu_{ij} = im - m + j$ & "m" is the max weight

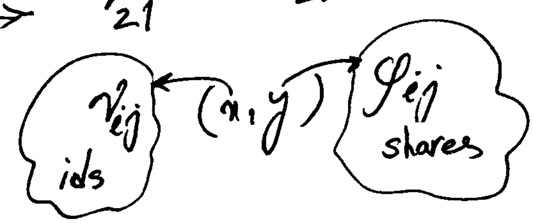
Example

$P_1 \longrightarrow \nu_{1j} = 1 \times 4 - 4 + j \xrightarrow{1 \leq j \leq 2} \nu_{11} = 1, \nu_{12} = 2$

$\boxed{m=4 \quad w_1 = 2}$

$P_2 \longrightarrow \nu_{2j} = 2 \times 4 - 4 + j \xrightarrow{1 \leq j \leq 3} \nu_{21} = 5, \nu_{2,2} = 6, \nu_{23} = 7$

$\boxed{m=4 \quad w_2 = 3}$



secret Recovery

If the total weight of participating players is more than t, use Lagrange interpolation to recover the secret

each P_i sends/reveals his shares s_{ij} for $1 \leq j \leq w_i$

$f(0) = \alpha$ defines the secret