Calcul multipartite sécurisé

Intervention dans le cours de cryptographie avancée de Matthieu Rambaud

Angelo Saadeh

Télécom Paris

10 mars 2022

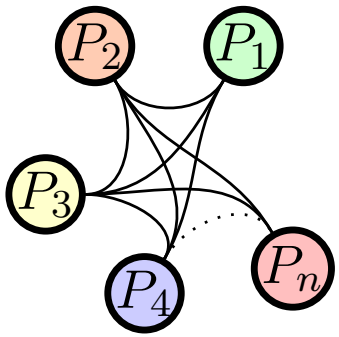
Motivation

Comment calculer la somme de plusieurs salaires sans les révéler ?

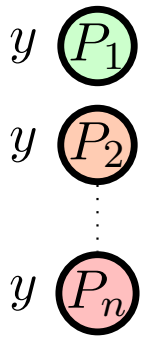
Plan

1. Introduction
   - Présentation du problème
   - Historique
2. Opérations de base
   - Partage de secret
   - Protocoles linéaires
   - Multiplication de secrets
   - Évaluation de fonctions
3. Sécurité
   - Types d'adversaires
4. Comparaisons
   - Utilisation des comparaisons
   - Exemples de protocoles de comparaison

Présentation du problème



$$y = f(s^1, \ldots, s^n) \longrightarrow$$



Définitions

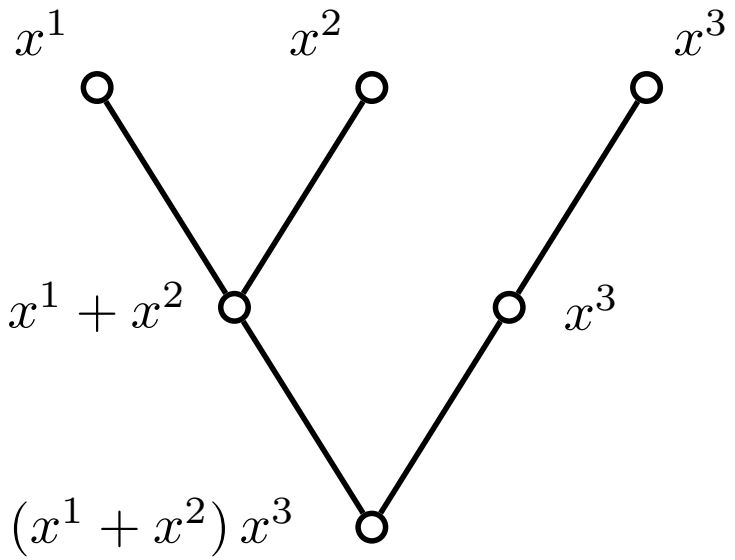
- **Secret** : donnée confidentielle détenue par un individu.
- **Joueurs** (ou parties) : les n individus qui détiennent les secrets et veulent calculer f sans les révéler.
- $\mathbb{Z}/q\mathbb{Z}$: corps fini (q premier) ou anneau fini (q puissance de 2) dans lequel se font les calculs.

Quels sont les cas d'usage ?

Historique

- 1982 : problème des millionnaires (Yao).
- 1986–1987 : circuits booléens (portes AND et XOR) : circuits brouillés de Yao, protocole GMW.
- 1988 : circuits arithmétiques sur un corps fini, à partir du partage de Shamir (BGW, CCD).

Exemple de circuit arithmétique à 3 joueurs, de secrets x^1, x^2, x^3 :



L'objectif est d'évaluer le circuit nœud par nœud, avec le protocole correspondant à l'opération de chaque nœud.

Partage de secret

Soient n le nombre de joueurs et $s \in \mathbb{Z}/q\mathbb{Z}$ un secret.

Définition : partage de secret à seuil (t, n)

Soit $t \in \mathbb{N}^*$, $t < n$. Un partage de secret à seuil (t, n) distribue s en n parts $s_1, \ldots, s_n \in \mathbb{Z}/q\mathbb{Z}$ telles que :

- t parts ou moins ne donnent aucune information sur s (t-confidentialité) ;
- $t + 1$ parts ou plus permettent de reconstruire s ($(t + 1)$-reconstruction).

Partage de secret

Soient n le nombre de joueurs et $s \in \mathbb{Z}/q\mathbb{Z}$ un secret.

Définition : partage de secret à seuil (t, n)

Soit $t \in \mathbb{N}^*$, $t < n$. Un partage de secret à seuil (t, n) distribue s en n parts $s_1, \ldots, s_n \in \mathbb{Z}/q\mathbb{Z}$ telles que :

- t parts ou moins ne donnent aucune information sur s (t-confidentialité) ;
- $t + 1$ parts ou plus permettent de reconstruire s ($(t + 1)$-reconstruction).

Exemples :

1. partage additif : seuil $(n - 1, n)$;
2. partage de Shamir : seuil (t, n).

Schémas de partage de secret

Partage de secret de Shamir

Entrée : un secret $s \in \mathbb{Z}/q\mathbb{Z}$, avec q premier et $q > n$. Sortie : n parts.

1. Le détenteur du secret tire un polynôme aléatoire f_s de degré au plus t tel que $f_s(0) = s$.
2. Il envoie $f_s(i)$ au joueur P_i, pour $i = 1, \ldots, n$.

Remarque : la multiplication de secrets demandera en plus $t < n/2$.

Partage de secret additif

Entrée : un secret $s \in \mathbb{Z}/q\mathbb{Z}$. Sortie : n parts.

1. Le détenteur du secret tire $s_1, \ldots, s_{n-1}$ uniformément dans $\mathbb{Z}/q\mathbb{Z}$ et pose $s_n = s - \sum_{i<n} s_i$, de sorte que $\sum_i s_i = s$.
2. Il envoie s_i au joueur P_i.

Addition de secrets

Addition : partage de Shamir

Entrée : chaque joueur P_i détient $x_i = f_x(i)$ et $y_i = f_y(i)$, où f_x, f_y sont de degré au plus t, avec $f_x(0) = x$ et $f_y(0) = y$. On note $[x]_t$ et $[y]_t$.
Sortie : $[x + y]_t$.

1. Chaque joueur P_i calcule $z_i \leftarrow x_i + y_i$, car $f_x + f_y$ est de degré au plus t et vaut $x + y$ en 0.

Addition : partage additif

Entrée : chaque joueur P_i détient x_i et y_i, avec $\sum_i x_i = x$ et $\sum_i y_i = y$. On note $[x]$ et $[y]$.
Sortie : $[x + y]$.

1. Chaque joueur P_i calcule $z_i \leftarrow x_i + y_i$.

Addition d'une constante publique α

Addition d'une constante : partage de Shamir

Entrée : $[x]_t$ et une constante publique α. Sortie : $[\alpha + x]_t$.

1. Chaque joueur P_i calcule $z_i \leftarrow x_i + \alpha$, car $f_x + \alpha$ vaut $x + \alpha$ en 0.

Addition d'une constante : partage additif

Entrée : $[x]$ et une constante publique α. Sortie : $[\alpha + x]$.

1. Chaque joueur P_i calcule $z_i \leftarrow \begin{cases} x_i + \alpha & \text{si } i = 1, \\ x_i & \text{sinon.} \end{cases}$

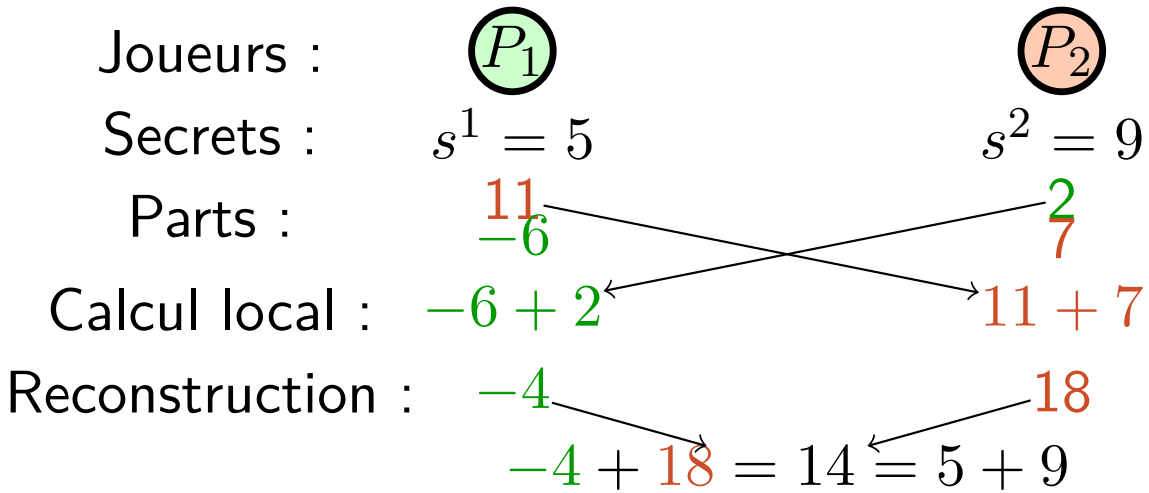
Un seul joueur ajoute la constante.

Protocoles linéaires et locaux

Les opérations suivantes se font localement, sans communication :

- addition de secrets ;
- multiplication par un scalaire public : chaque joueur multiplie sa part par la constante ;
- addition d'une constante publique.

Exemple : somme de deux secrets par partage additif.



Multiplication de secrets : partage de Shamir

Interpolation de Lagrange

Soient q premier, f un polynôme de degré au plus d et $C \subset \mathbb{Z}/q\mathbb{Z}$ avec $|C| = d+1$.

$$f(X) = \sum_{i \in C} f(i)\,\delta_i(X), \qquad \delta_i(X) = \prod_{\substack{j \in C \\ j \neq i}} \frac{X-j}{i-j}.$$

Soient $t < n/2$, $C \subseteq \{1, \ldots, n\}$ avec $|C| = 2t+1$, et $r_i = \delta_i(0)$.

Multiplication de secrets : partage de Shamir

Interpolation de Lagrange

Soient q premier, f un polynôme de degré au plus d et $C \subset \mathbb{Z}/q\mathbb{Z}$ avec $|C| = d+1$.

$$f(X) = \sum_{i \in C} f(i)\,\delta_i(X), \qquad \delta_i(X) = \prod_{\substack{j \in C \\ j \neq i}} \frac{X-j}{i-j}.$$

Soient $t < n/2$, $C \subseteq \{1, \ldots, n\}$ avec $|C| = 2t+1$, et $r_i = \delta_i(0)$.

Protocole de multiplication

Entrée : $[x]_t$, $[y]_t$. Sortie : $[xy]_t$.

1. Chaque joueur P_i calcule $z_i = x_i y_i = (f_x f_y)(i)$: les z_i forment $[xy]_{2t}$.
2. Chaque joueur P_i, $i \in C$, partage z_i : $[z_i]_t$.
3. Chaque joueur P_j calcule localement $\sum_{i \in C} r_i\,[z_i]_t$.

Correction : $\sum_{i \in C} r_i z_i = \sum_{i \in C} r_i (f_x f_y)(i) = (f_x f_y)(0) = xy$, et une combinaison linéaire de partages de degré t est un partage de degré t.

Multiplication de secrets : partage additif (triplets de Beaver)

Protocole de multiplication

Entrée : $[x]$, $[y]$. Sortie : $[xy]$.

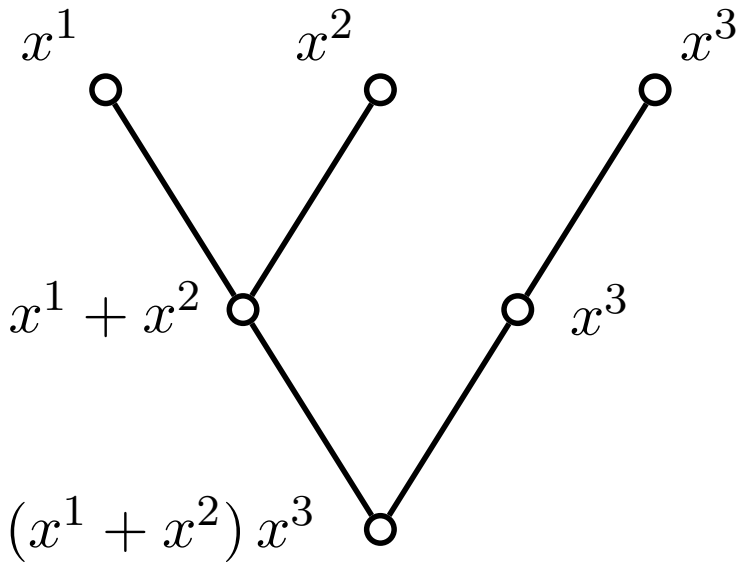
1. Les joueurs reçoivent au préalable un triplet $[a], [b], [c]$, avec a, b uniformes dans $\mathbb{Z}/q\mathbb{Z}$ et $c = ab$.
2. Chaque joueur P_i calcule $d_i = x_i - a_i$ et $e_i = y_i - b_i$: on obtient $[d]$ et $[e]$.
3. Les joueurs ouvrent d et e (ils ne révèlent rien, car a et b sont uniformes).
4. Les joueurs calculent $[z] = [c] + d\,[b] + e\,[a] + de$ (la constante de est ajoutée par un seul joueur).

Correction :

$$\begin{aligned} z &= ab + (x-a)\,b + (y-b)\,a + (x-a)(y-b) \\ &= ab + xb - ab + ya - ab + xy - xb - ay + ab \;=\; xy. \end{aligned}$$

Évaluation de fonctions

Évaluer un circuit arithmétique (portes d'addition et de multiplication, puis reconstruction) :



1. Partager les secrets avec un schéma de partage de secret.
2. Évaluer chaque nœud du circuit avec le protocole correspondant.
3. Reconstruire le résultat final.

Types d'adversaires

Adversaires

- **Passifs** (honnêtes mais curieux) : ils suivent le protocole, mais cherchent à apprendre des informations à partir des messages reçus.
- **Actifs** (malveillants) : ils peuvent s'écarter du protocole, par exemple envoyer de faux messages.

Utilisation des comparaisons

Les comparaisons ne sont pas des opérations arithmétiques : en MPC, on ne peut pas lire directement le bit de poids fort d'un secret partagé.

Protocole de comparaison $\Pi(x, R)$

Entrée : $[x]$, avec $x \in [0, q)$, et une constante publique $R \in [0, q)$.

Sortie : $\begin{cases} [1] & \text{si } x > R, \\ [0] & \text{sinon.} \end{cases}$

Elles servent à :

- obtenir le signe ;
- calculer des fonctions par morceaux ;
- calculer la valeur absolue ;
- calculer des maximums et des minimums.

Exemples de protocoles de comparaison

- edaBits (Escudero et al., CRYPTO 2020) : variantes probabilistes ou déterministes.
- Partage de secret de fonctions, FSS (Boyle, Gilboa, Ishai) : phase en ligne rapide.
- Rabbit (Makri et al., 2021) : comparaison efficace par conversion arithmétique-binaire.

Exercices

Exercice 1. À partir d'un protocole de comparaison $\Pi(x, R)$, construire un protocole qui retourne $\begin{cases} [x] & \text{si } x > R, \\ [2x] & \text{sinon.} \end{cases}$

Exercice 2. À partir d'un protocole de comparaison $\Pi(x, R)$, construire un protocole qui retourne, pour $R < R'$, $\begin{cases} [0] & \text{si } x < R, \\ [x] & \text{si } R < x < R', \\ [1] & \text{si } x > R'. \end{cases}$