Cryptographie Avancée

Introduction au Calcul MultiPartite (Sécurisé)

Maxime Bombar

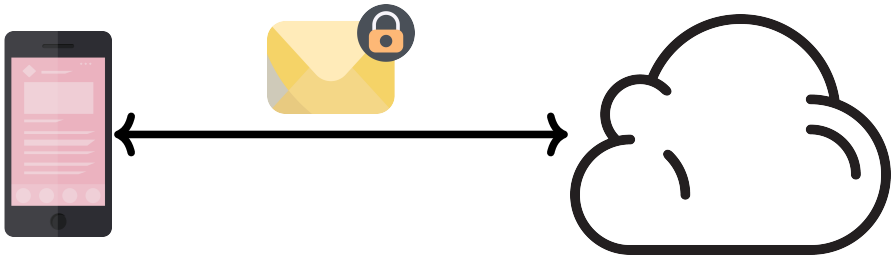
Vendredi 15 Novembre

Plan

- Vendredi 15 Novembre - Cours 01 : MPC Inconditionnel et Partage de Secrets
- Lundi 18 Novembre - Cours 02 : Extensions sous Hypothèses Calculatoires
- Lundi 25 Novembre - Cours 03 : MPC-in-the-Head et Preuves Zéro-Knowledge.

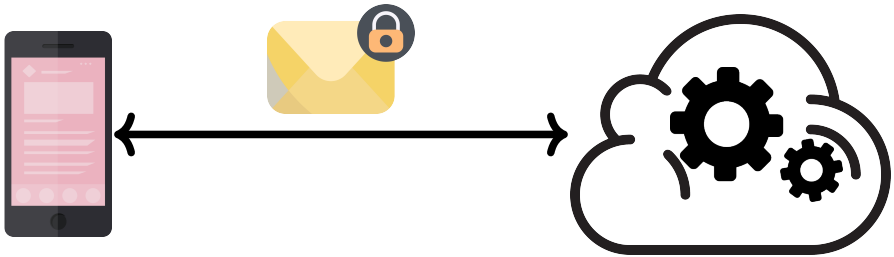
Introduction

Calcul Sécurisé



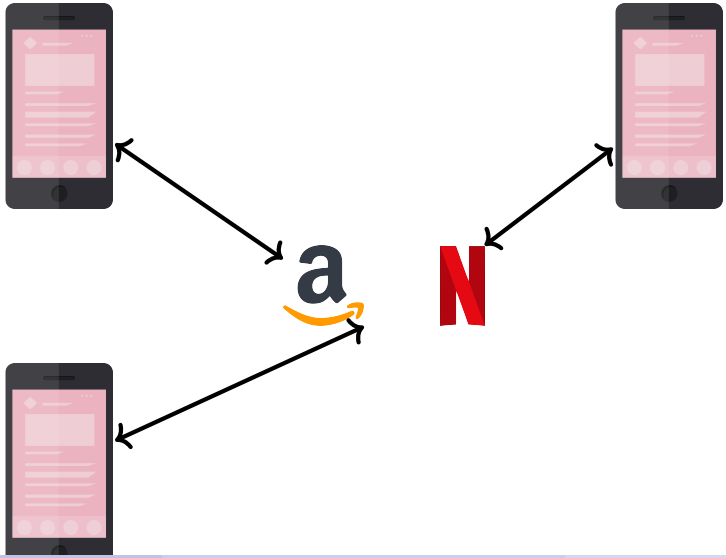
- **Cryptographie traditionnelle** : Protection des données statiques, ou en transit.

Calcul Sécurisé

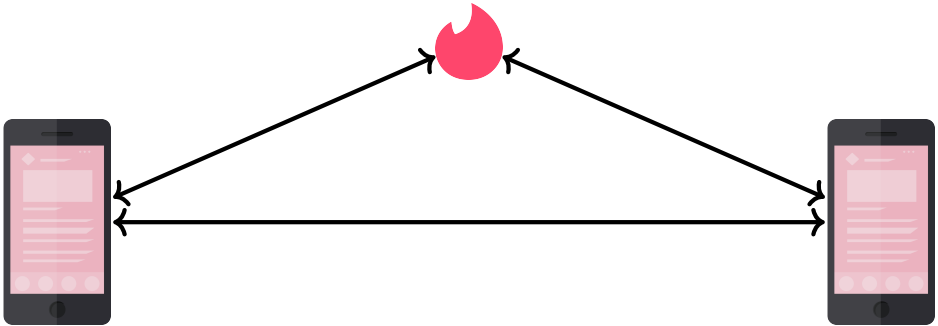


- **Cryptographie traditionnelle** : Protection des données statiques, ou en transit.
- Sécurité de nos **calculs** ?

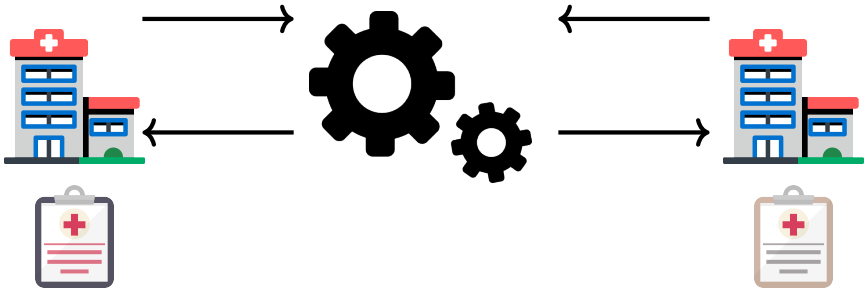
Exemple : Système de Recommandation



Exemple : Applications de Rencontre



Exemple : Données de Santé



Vous avez dit « Calcul Sécurisé » ?

La Problématique : Est-il possible de réaliser un calcul sur des données distribuées à plusieurs parties, sans que celles-ci ne soient divulguées ?

Propriétés Voulues :

- **Correctness** : Tout le monde reçoit le résultat du calcul.
- **Privacy** : Les parties n'apprennent rien de plus que ce qu'ils peuvent déduire de leur entrée et du résultat.

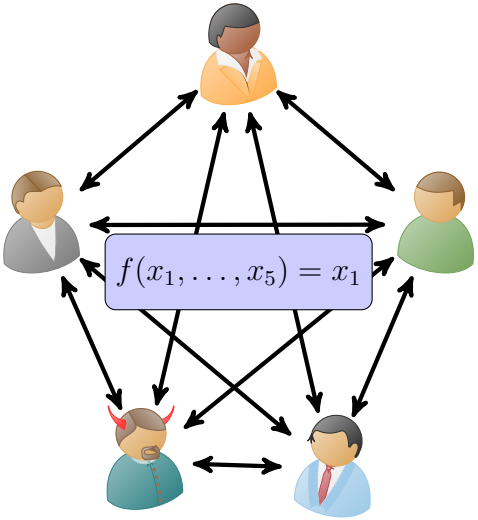
Vous avez dit ≪ Calcul Sécurisé ≫ ?

La Problématique : Est-il possible de réaliser un calcul sur des données distribuées à plusieurs parties, sans que celles-ci ne soient divulguées ?

Hypothèses :

- Les communications sont authentifiées.
- Communications soit en broadcast, soit entre deux parties.

Qu'a-t-on le droit d'apprendre ?



- Si le protocole est correct, toutes les parties apprennent x_1.
- La propriété de **privacy** ne s'applique qu'aux éléments qui ne peuvent pas être déduites du résultat!

Quelques Définitions

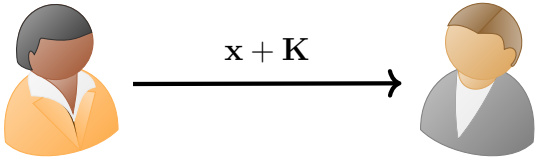
Un **protocole** est une suite d'instructions entre plusieurs acteurs, qui précisent à chaque étape quelle opération est réalisée, et par quelle partie.

Un adversaire est modélisé par un sous ensemble de parties appelées **parties corrompues**, qu'il contrôle totalement. L'adversaire est dit **semi-honnête** si les parties corrompues respectent le protocole, et **malicieux** dans le cas contraire.

Comme d'habitude, on distinguera le cas où l'adversaire a une puissance de calcul infini (**sécurité parfaite**, ou inconditionnelle), du cas où l'adversaire a des ressources limitées (**sécurité calculatoire**).

Rappel : Sécurité Parfaite

$\mathbf{x} \in G$ (Groupe fini)
$\mathbf{K} \leftarrow G$ (Aléatoire)



La distribution de $\mathbf{x} + \mathbf{K}$ est **indépendante** de $\mathbf{x}$.

Un Premier Protocole

- Les joueurs possèdent chacun une entrée $x_i \in \mathbb{Z}$.
- Ils veulent calculer $\sum_i x_i$.

Comment faire ?

Un Premier Protocole

x_5

x_4

- Les joueurs possèdent chacun une entrée $x_i \in \mathbb{Z}$.
- Ils veulent calculer $\sum_i x_i$.

x_1

$x_1 + K$

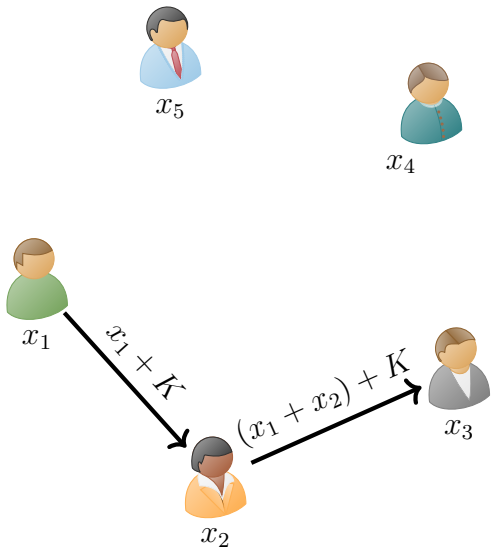
x_3

x_2

Solution :

- P_1 choisit $K \leftarrow \mathbb{Z}/N\mathbb{Z}$ pour N assez grand et envoie $y_1 \stackrel{\text{def}}{=} x_1 + K \mod N$ (**OTP**).

Un Premier Protocole

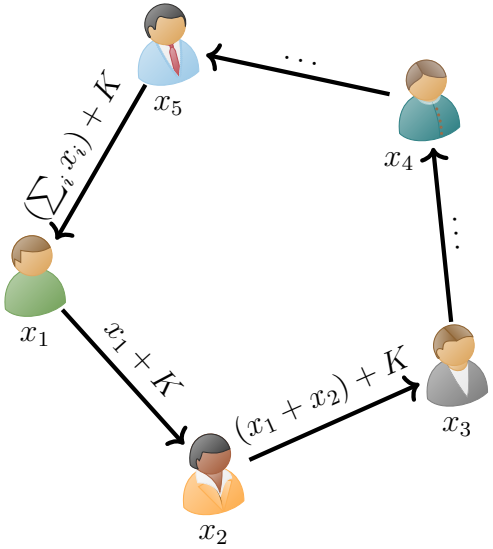


- Les joueurs possèdent chacun une entrée $x_i \in \mathbb{Z}$.
- Ils veulent calculer $\sum_i x_i$.

Solution :

- P_1 choisit $K \leftarrow \mathbb{Z}/N\mathbb{Z}$ pour N assez grand et envoie $y_1 \stackrel{\text{def}}{=} x_1 + K \mod N$ (**OTP**).
- P_2 calcule $y_1 + x_2 = (x_1 + x_2) + K \mod N$.

Un Premier Protocole

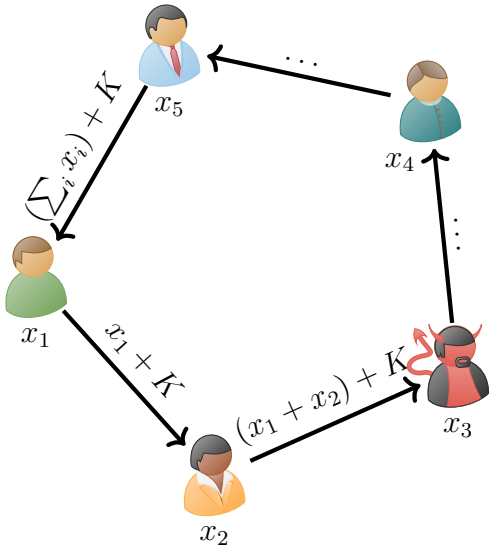


- Les joueurs possèdent chacun une entrée $x_i \in \mathbb{Z}$.
- Ils veulent calculer $\sum_i x_i$.

Solution :

- P_1 choisit $K \leftarrow \mathbb{Z}/N\mathbb{Z}$ pour N assez grand et envoie $y_1 \stackrel{\text{def}}{=} x_1 + K \mod N$(**OTP**).
- P_2 calcule $y_1 + x_2 = (x_1 + x_2) + K \mod N$.
- $\ldots P_1$ récupère $(\sum x_i) + K \mod N$, retire K et broadcast le résultat.

Un Premier Protocole

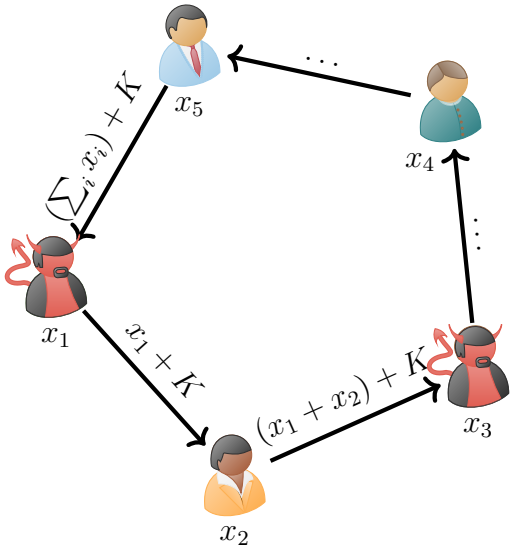


- Les joueurs possèdent chacun une entrée $x_i \in \mathbb{Z}$.
- Ils veulent calculer $\sum_i x_i$.

Sécurité :

- Si une seule personne est corrompue, elle n'apprend rien.

Un Premier Protocole



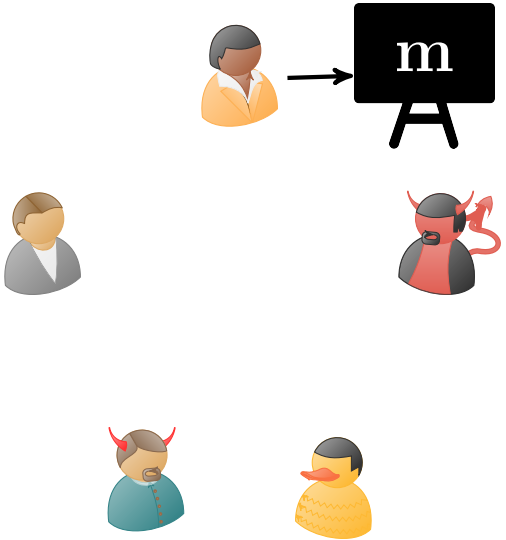
- Les joueurs possèdent chacun une entrée $x_i \in \mathbb{Z}$.
- Ils veulent calculer $\sum_i x_i$.

Sécurité :

- Si une seule personne est corrompue, elle n'apprend rien.
- Si P_1 et P_3 sont corrompus, P_3 peut apprendre x_2 !

Définir et Prouver la Sécurité

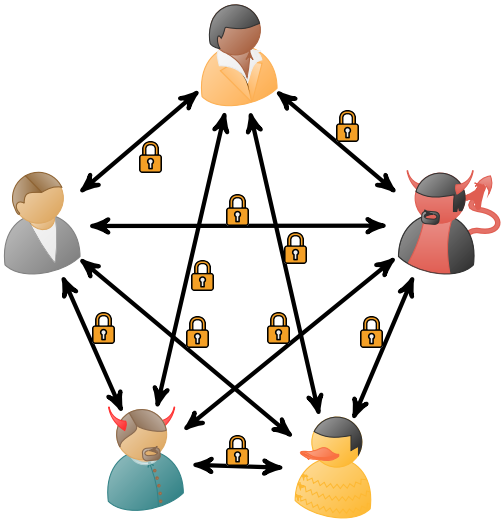
Le modèle de Communication



Toutes les communications sont authentifiées

- **Broadcast :**
  Chaque partie peut écrire son message sur un tableau publique qui est lu par tout le monde.

Le modèle de Communication

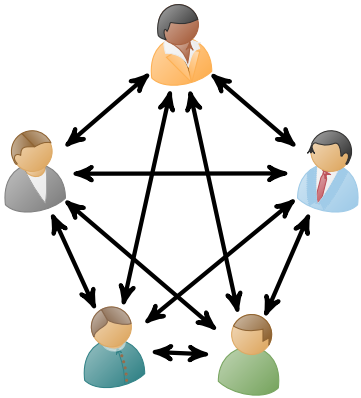


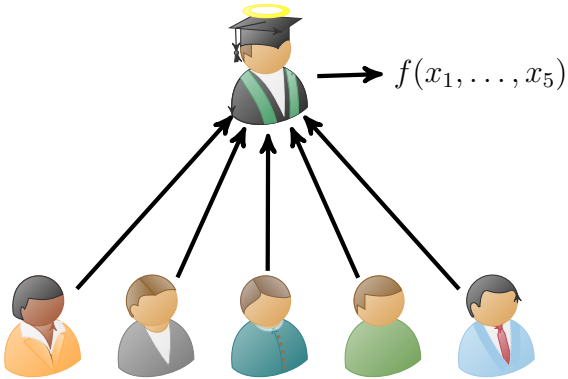
Toutes les communications sont authentifiées

- **Broadcast :**
  Chaque partie peut écrire son message sur un tableau publique qui est lu par tout le monde.

- **Canaux pair-à-pair :**
  Chaque paire de parties est connectée par un cannal sécurisé et authentifié.

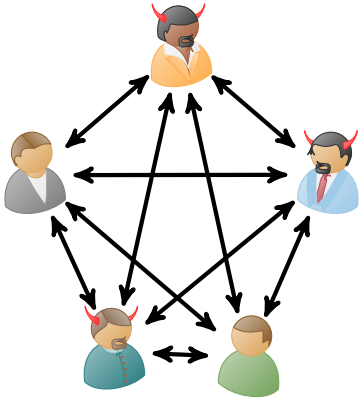
Modèle Idéal

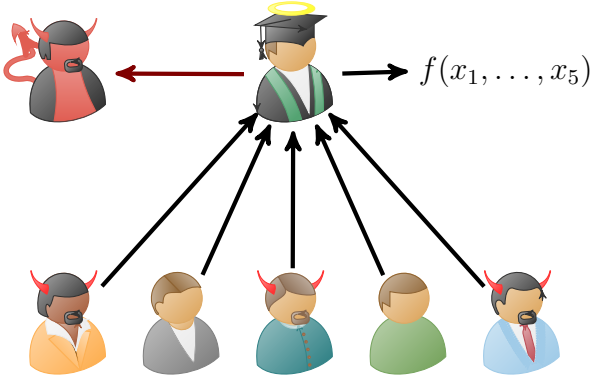




Pour étudier la sécurité, on modélise le protocole par un tiers de confiance (parfois aussi appelé la **fonctionnalité idéale**) qui reçoit toutes les entrées et calcule le résultat.

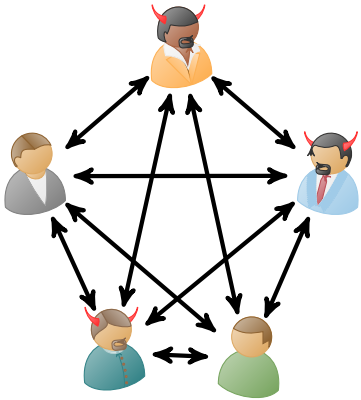
Modèle Idéal

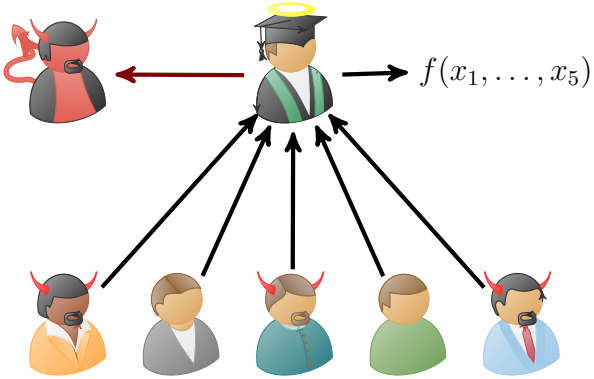




Dans le modèle idéal, on modélise l'adversaire par une fuite d'information correspondant à l'ensemble des objets qu'il peut voir au travers des parties corrompues.

Modèle Idéal





Simulateur : Formellement, on construit dans le monde idéal un algorithme qu'on appelle un **simulateur** qui émule, tout ce que peut voir l'adversaire au travers du protocole réél.

Vue d'un joueur

Définition : Pour un protocole π à n joueurs $P_1, \ldots, P_n$, on définit la **vue** view_j du joueur P_j comme l'ensemble des valeurs que voit P_j durant l'exécution du protocole : L'entrée de P_j, toutes les valeurs reçues par P_j ainsi que toutes les valeurs aléatoires qu'il génère.

Dans le protocole simple de tout à l'heure, pour $j > 1$, la vue de P_j était

$$\mathrm{view}_j \overset{\text{def}}{=} \left\{ x_j, s_j \overset{\text{def}}{=} \sum_{i=1}^{j-1} x_i + K, \sum_{i=1}^{n} x_i \right\}$$

où K était uniformément distribué dans $\mathbb{Z}/N\mathbb{Z}$.

Sécurité Passive et Simulateur

Définition : Soit π un protocole à n joueurs $P_1, \ldots, P_n$. On note $\mathscr{C}$ l'ensemble des parties corrompues. On note x_j (resp. y_j) l'entrée de P_j (resp. le résultat qu'il reçoit). Le protocole est dit **sécurisé** en présence de $\mathscr{C}$ s'il existe un algorithme probabiliste efficace (qu'on appelle **simulateur**) ayant accès à $\{(x_j, y_j)\}_{P_j \in \mathscr{C}}$ et qui produit un ensemble $\mathcal{S}$ **indistinguable** de $\{\mathrm{view}_j \mid j \in \mathscr{C}\}$.

Dans le protocole simple de tout à l'heure, on peut simuler view_j en tirant uniformément aléatoirement un élément de $\mathbb{Z}/N\mathbb{Z}$ pour remplacer s_j.

Sécurité Passive et Simulateur

Définition : Soit π un protocole à n joueurs $P_1, \ldots, P_n$. On note $\mathscr{C}$ l'ensemble des parties corrompues. On note x_j (resp. y_j) l'entrée de P_j (resp. le résultat qu'il reçoit). Le protocole est dit **sécurisé** en présence de $\mathscr{C}$ s'il existe un algorithme probabiliste efficace (qu'on appelle **simulateur**) ayant accès à $\{(x_j, y_j)\}_{P_j \in \mathscr{C}}$ et qui produit un ensemble $\mathcal{S}$ **indistinguable** de $\{\mathrm{view}_j \mid j \in \mathscr{C}\}$.

Dans le protocole simple de tout à l'heure, on peut simuler view_j en tirant uniformément aléatoirement un élément de $\mathbb{Z}/N\mathbb{Z}$ pour remplacer s_j.

Question : Que signifie **indistinguable** ?

Rappel : Distance Statistique

Soient X et Y deux variables aléatoires à valeurs dans un ensemble fini $\mathcal{E}$. Leur distance statistique (ou distance en variation totale) est définie par

$$\Delta(X,Y) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{a\in\mathcal{E}} \Big|\mathbb{P}(X=a) - \mathbb{P}(Y=a)\Big|.$$

Data Processing Inequality : Soit f une fonction éventuellement aléatoire, mais telle que l'aléas interne de f soit indépendante de X et Y. Alors

$$\Delta(f(X), f(Y)) \leqslant \Delta(X,Y).$$

Rappel : Indistinguabilité Statistique

Deux variables aléatoires $X(n)$ et $Y(n)$, indéxées par un certain paramètre n, sont dites **statistiquement indistinguables** si leur distance statistique est négligeable :

$$\Delta(X(n), Y(n)) = \mathrm{negl}(n).$$

En pratique, on peut souvent même imposer une distance statistique nulle, comme dans notre exemple : la vue simulée a **exactement la même** distribution que la vue réelle du joueur corrompu.

Rappel : Indistingabilité Calculatoire

Soient $\mathcal{D}_0$ et $\mathcal{D}_1$ deux distributions définies sur un même espace $\mathcal{E}$, et soit $\mathcal{A}$ un algorithme qui prend en entrée un élément de $\mathcal{E}$ et sort un bit b. On définit l'**avantage** de $\mathcal{A}$ par

$$\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1) \stackrel{\text{def}}{=} \left| \mathbb{P}\left(\mathcal{A}(X) = 1 \mid X \leftarrow \mathcal{D}_1\right) - \mathbb{P}\left(\mathcal{A}(X) = 1 \mid X \leftarrow \mathcal{D}_0\right) \right|.$$

Deux distributions $\mathcal{D}_0(n)$ et $\mathcal{D}_1(n)$ sont **calculatoirement indistinguables** si pour tout PPT $\mathcal{A}$,

$$\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0(n), \mathcal{D}_1(n)) = \mathrm{negl}(n).$$

En général on note $\mathcal{D}_0 \approx_c \mathcal{D}_1$.

Avantage et Distance Statistique

Remarque : La distance statistique entre deux distributions $\mathcal{D}_0$ et $\mathcal{D}_1$ est l'avantage maximal d'un distingueur $\mathcal{A}$ sans hypothèse de ressources limitées.

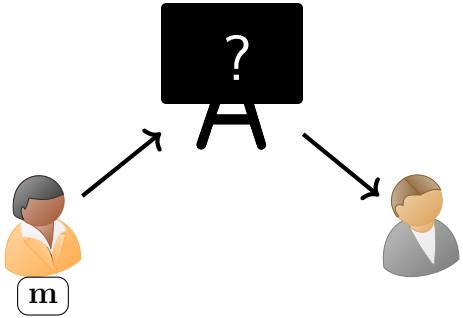
Sécurité Passive

Un protocole π à n joueurs $P_1, \ldots, P_n$ réalise une fonctionnalité $\mathcal{F}$ en présence d'au plus t adversaires **semi-honnêtes** s'il existe un simulateur Sim tel que pour chaque sous ensemble de parties $\mathscr{C}$ de cardinal au plus t, la distribution

$$\mathrm{Sim}((x_j, \mathcal{F}_j)_{P_j \in \mathcal{C}}) \approx_c \{\mathrm{view}_{j,\pi} \mid P_j \in \mathscr{C}\}.$$

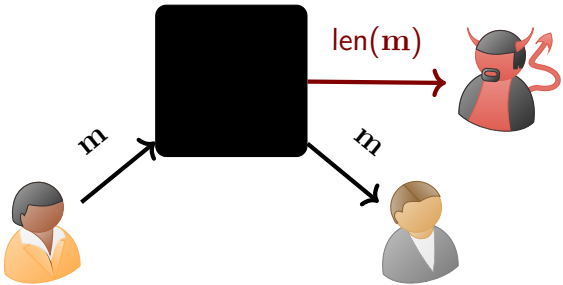
La sécurité est dite **parfaite** si les deux distributions sont en réalité **statistiquement indistinguables**.

Exemple : Communication Sécurisée



Monde réel

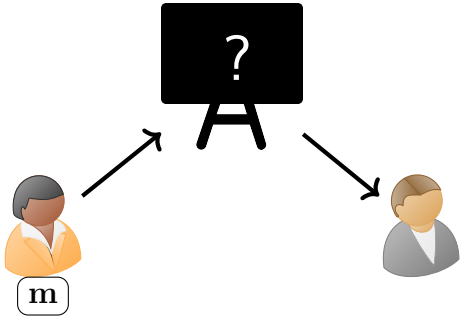
Attaquant peut voir tous les messages envoyés par Alice, mais ne doit rien apprendre sur $\mathbf{m}$ d'autre qu'éventuellement sa taille.



Monde Idéal

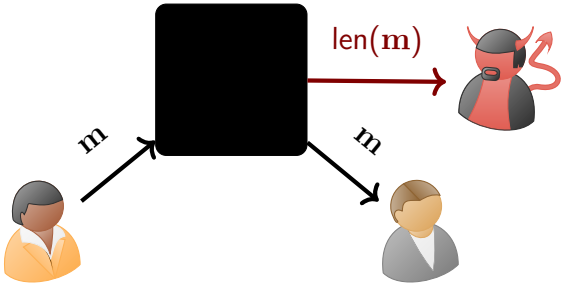
Comment modéliser ce protocole ?

Exemple : Communication Sécurisée



Monde réel

Attaquant peut voir tous les messages envoyés par Alice, mais ne doit rien apprendre sur $\mathbf{m}$ d'autre qu'éventuellement sa taille.



Monde Idéal

Comment modéliser ce protocole ?

Chiffrement IND-CPA!

Un Résultat de Composition

Définition : Un protocole π calculant une fonctionnalité de la forme $f \circ g$ est dit étudié dans le modèle g-hybride si tous les appels à g sont remplacés par une fonctionnalité idéale calculant g.

Théorème (Canetti 2000) : Soit π un protocole calculant une fonctionnalité $\mathcal{F}$ et prouvé sûr contre t adversaires dans le modèle g-hybride. Soit σ un protocole sûr face à t adversaires, et calculant g. Si π réalise un seul appel à g par tour alors le protocole $\pi[\sigma]$ obtenu en remplaçant les appels à la fonctionnalité idéale $\}$ par le protocole σ calcule la fonctionnalité $\mathcal{F}$ et est sûr contre t-adversaires.

En général, pour prouver la sécurité d'un protocole compliqué π, on le découpe en protocoles beaucoup plus simples dont on prouve la sécurité un par un.

Rappel : Secret Sharing

Partage de Secret

Soit $\mathcal{S}$ l'ensemble des secrets et $\mathcal{P}$ l'ensemble des parts. Un schéma de partage de secrets à n joueurs et à seuil t est une paire d'algorithmes $(\mathrm{Shr}, \mathrm{Rec})$ tels que

$$\mathrm{Shr} : \mathcal{S} \to \mathcal{P}^n \quad \mathrm{Rec} \colon \mathcal{P}^\star \to \mathcal{S}$$

et vérifiant

- **Correctness :** Si $(p_1, \ldots, p_n) = \mathrm{Shr}(s)$ alors

$$\mathbb{P}\left(\mathrm{Rec}(p_{i_1}, \ldots, p_{i_\ell}) = s\right) = 1 \qquad \forall \ell \geqslant t.$$

- **Perfect Privacy :** Soit $s_1, s_2 \in \mathcal{S}$ alors les distributions obtenues en extrayant $k < t$ éléments de $\mathrm{Shr}(s_1)$ et $\mathrm{Shr}(s_2)$ sont statistiquement indistinguables.

(les probabilités sont prises sur l'aléas interne des algorithmes).

Exemple : $(n, n)-$Secret Sharing

Soit G un groupe fini, par exemple $\mathbb{Z}/N\mathbb{Z}$, ou $\mathbb{F}_q^\ell$.

Exemple : $(n, n)-$Secret Sharing

Soit G un groupe fini, par exemple $\mathbb{Z}/N\mathbb{Z}$, ou $\mathbb{F}_q^\ell$.

Pour $x \in G$, l'algorithme Shr tire uniformément $p_1, \ldots, p_{n-1} \leftarrow G$, calcule

$$p_n \stackrel{\text{def}}{=} x - \sum_{i=1}^{n-1} p_i.$$

et renvoie $(p_1, \ldots, p_n)$.

Exemple : $(n, n)-$Secret Sharing

Soit G un groupe fini, par exemple $\mathbb{Z}/N\mathbb{Z}$, ou $\mathbb{F}_q^\ell$.

Pour $x \in G$, l'algorithme Shr tire uniformément $p_1, \ldots, p_{n-1} \leftarrow G$, calcule

$$p_n \stackrel{\text{def}}{=} x - \sum_{i=1}^{n-1} p_i.$$

et renvoie $(p_1, \ldots, p_n)$.

L'algorithme Rec calcule simplement la somme de ses entrées.

- Sur l'entrée $(p_1, \ldots, p_n)$ il retrouve x.

Exemple : $(n, n)-$Secret Sharing

Soit G un groupe fini, par exemple $\mathbb{Z}/N\mathbb{Z}$, ou $\mathbb{F}_q^\ell$.

Pour $x \in G$, l'algorithme Shr tire uniformément $p_1, \ldots, p_{n-1} \leftarrow G$, calcule

$$p_n \stackrel{\text{def}}{=} x - \sum_{i=1}^{n-1} p_i.$$

et renvoie $(p_1, \ldots, p_n)$.

L'algorithme Rec calcule simplement la somme de ses entrées.

- Sur l'entrée $(p_1, \ldots, p_n)$ il retrouve x.
- Avec $k < n$ parts, la somme est **uniformément distribuée** dans G.

Shamir (k, n)-Secret Sharing

On se place sur un corps fini $\mathbb{F}_q$ avec $q > n$.

Shamir (k, n)-Secret Sharing

On se place sur un corps fini $\mathbb{F}_q$ avec $q > n$.

- Pour $s \in \mathbb{F}_q$, l'algorithme Shr sélectionne $a_1, \ldots, a_{k-1}$ uniformément aléatoirement dans $\mathbb{F}_q$ et définit $A \stackrel{\text{def}}{=} s + a_1 X + \cdots + a_{k-1} X^{k-1}$ ainsi que $x_1, \ldots, x_n$ des éléments distincts de $\mathbb{F}_q$.
- La part de P_i est définie comme $(x_i, A(x_i))$.

Shamir (k, n)-Secret Sharing

On se place sur un corps fini $\mathbb{F}_q$ avec $q > n$.

- Pour $s \in \mathbb{F}_q$, l'algorithme Shr sélectionne $a_1, \ldots, a_{k-1}$ uniformément aléatoirement dans $\mathbb{F}_q$ et définit $A \stackrel{\text{def}}{=} s + a_1 X + \cdots + a_{k-1} X^{k-1}$ ainsi que $x_1, \ldots, x_n$ des éléments distincts de $\mathbb{F}_q$.
- La part de P_i est définie comme $(x_i, A(x_i))$.

L'algorithme Rec procède alors à une **interpolation de Lagrange** sur k parts pour retrouver A, puis l'évalue en 0.

Le Protocole BGW

MPC Parfait avec Sécurité Passive

Le protocole de Ben-Or, Goldwasser et Wigderson (1988)

Pour n'importe quelle fonction $f(x_1, \ldots, x_n)$, il existe un protocole de MPC à n joueurs, capable de calculer f avec **Sécurité Parfaite**, en présence d'un adversaire **semi-honnête**, et controlant jusqu'à $\mathbf{t < n/2}$ parties.

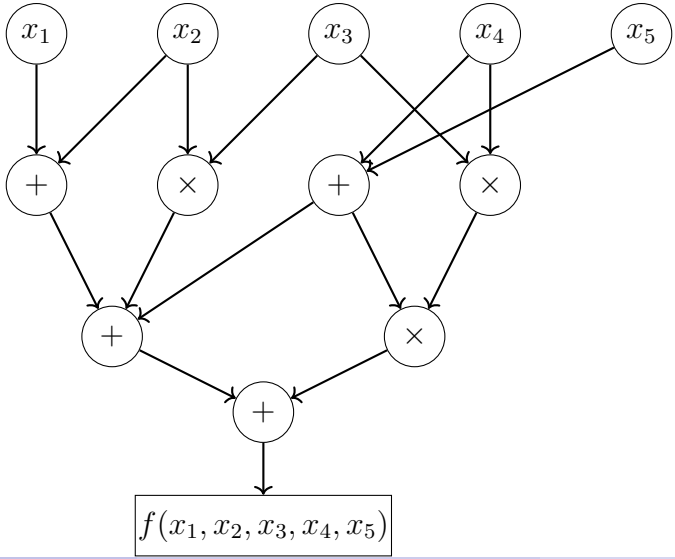
- La fonction $f : \mathbb{F}^n \to \mathbb{F}^n$ est représentée sous la forme d'un circuit arithmétique.
- On réalise un partage de secrets des entrées.
- On réalise des protocoles de calcul élémentaires au niveau de chaque porte.

Circuits Arithmétiques

Un circuit arithmétique est un graphe orienté acyclique dont les noeuds internes sont appelés **portes** (*gates*) et les arêtes sont appelés **cables** (*wires*). Chaque porte a au plus 2 cables en entrée.

- Il existe n portes avec 0 cables en entrée, et 1 sortie. Ils correspondent aux entrées secrètes des P_i.
- Chaque porte interne représente soit une *addition* ou une *multiplication* (deux cables en entrée, autant de sortie que souhaitée), ou encore une *multiplication par une constante* (un cable en entrée).

Évaluation de Circuit

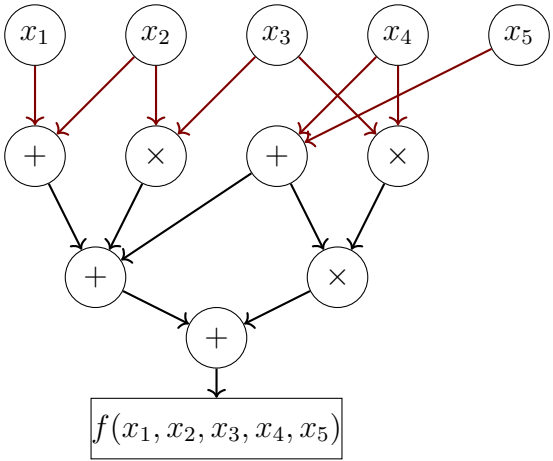


Le Protocole BGW - Idées

Chaque entrée du protocole n'est connue que d'une seule partie. Les valeurs intermédiaires ne doivent révéler aucune information sur les x_i.
Les parties vont chacun **émuler** le calcul du circuit localement le plus possible.

Invariant : Les valeurs sur chaque cable sont des $(t+1, n)-$Secret Sharing des valeurs correspondant au circuit évalué sur toutes les entrées.

Le Protocole BGW - Étape I : Partage des Entrées

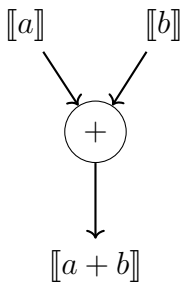


Chaque partie P_i réalise un $(t+1, n)$-Secret Sharing de son entrée x_i et distribue la part $[\![x_i; A_i]\!]_j$ au joueur P_j.

Hypothèse : $t < n/2$.

Le Protocole BGW - Étape II-a : Addition

Comment passer une porte additive ?



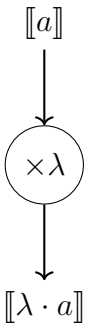
Invariant : Il existe des polynômes Q_a et Q_b tels que

- $\deg(Q_a) \leqslant t$, $\deg(Q_b) \leqslant t$
- $Q_a(0) = a$, $Q_b(0) = b$
- Chaque P_i connaît $Q_a(\alpha_i)$ et $Q_b(\alpha_i)$.

- On pose $Q_{a+b} \stackrel{\text{def}}{=} Q_a + Q_b$ qui est bien de degré $\leqslant t$ et vérifie $Q_{a+b}(0) = a + b$.
- Chaque partie peut calculer **localement** $Q_a(\alpha_i) + Q_b(\alpha_i) = Q_{a+b}(\alpha_i)$.

Le Protocole BGW - Étape II-b : Multiplication Publique

Comment passer une porte $\times\lambda$?



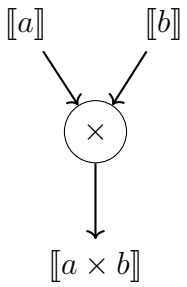
Invariant : Il existe un polynôme Q_a tel que

- $\deg(Q_a) \leqslant t$;
- $Q_a(0) = a$;
- Chaque P_i connaît $Q_a(\alpha_i)$.

- On pose $Q_{\lambda \cdot a} \stackrel{\text{def}}{=} \lambda \cdot Q_a$ qui est bien de degré $\leqslant t$ et vérifie $Q_{\lambda \cdot a}(0) = a$
- Chaque partie peut calculer **localement** $Q_a(\alpha_i) \times \lambda = Q_{\lambda \cdot a}(\alpha_i)$.

Le Protocole BGW - Étape II-c : Multiplication

Comment passer une porte multiplicative ?



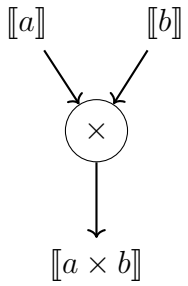
Invariant : Il existe des polynômes Q_a et Q_b tels que

- $\deg(Q_a) \leqslant t$, $\deg(Q_b) \leqslant t$
- $Q_a(0) = a$, $Q_b(0) = b$
- Chaque P_i connaît $Q_a(\alpha_i)$ et $Q_b(\alpha_i)$.

- On a bien $(Q_a \times Q_b)(0) = a \times b$.
- Chaque partie peut calculer **localement** $Q_a(\alpha_i) \times Q_b(\alpha_i)$.

Le Protocole BGW - Étape II-c : Multiplication

Comment passer une porte multiplicative ?



Invariant : Il existe des polynômes Q_a et Q_b tels que

- $\deg(Q_a) \leqslant t$, $\deg(Q_b) \leqslant t$
- $Q_a(0) = a$, $Q_b(0) = b$
- Chaque P_i connaît $Q_a(\alpha_i)$ et $Q_b(\alpha_i)$.

- On a bien $(Q_a \times Q_b)(0) = a \times b$.
- Chaque partie peut calculer **localement** $Q_a(\alpha_i) \times Q_b(\alpha_i)$.
- **Attention au degré!** $\deg(Q_a \times Q_b) \leqslant 2t$.

Conséquence de l'Interpolation de Lagrange

Hypothèse : H est un polynôme de degré $\leqslant n-1$.

Soit $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ deux à deux distincts, et $L_i(X)$ la base de Lagrange associée. Alors H s'écrit

$$H(X) = \sum_{i=1}^{n} H(\alpha_i) \cdot L_i(X).$$

En particulier,

$$H(0) = \sum_{i=1}^{n} H(\alpha_i) \cdot \underbrace{L_i(0)}_{\text{Indépendant de } H} = \sum_{i=1}^{n} H(\alpha_i) \cdot \underbrace{\boldsymbol{\lambda_i}}_{\text{Constantes Publiques}}$$

Multiplication et Réduction de Degré

Rappel : $t < n/2$.

On pose $H \stackrel{\text{def}}{=} Q_a \times Q_b$, de degré $\leqslant 2t \leqslant n-1$.

- P_i peut **localement** calculer $H(\alpha_i) = Q_a(\alpha_i) \times Q_b(\alpha_i)$.
- P_i choisit aléatoirement un polynôme H_i de degré $\leqslant t$ tel que $H_i(0) = H(\alpha_i)$ et distribue $H_i(\alpha_j)$ à P_j.

Après cette phase, chaque partie P_j possède $H_1(\alpha_j), \ldots, H_n(\alpha_j)$ et peut calculer

$$\boldsymbol{\lambda}_1 H_1(\alpha_j) + \cdots + \boldsymbol{\lambda}_n H_n(\alpha_j).$$

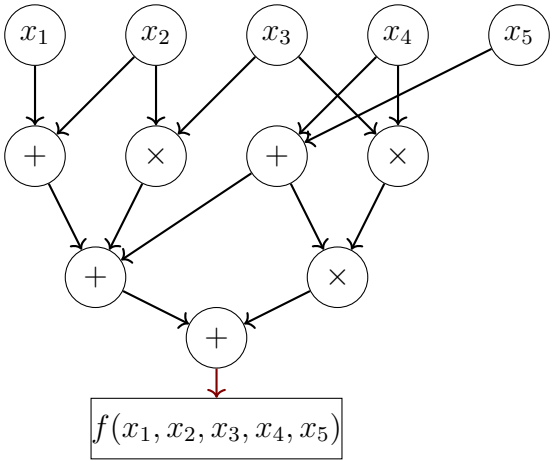
Porte Multiplicative (Preuve de Correction)

Soit $\Theta(X) \stackrel{\text{def}}{=} \boldsymbol{\lambda}_1 H_1(X) + \cdots + \boldsymbol{\lambda}_n H_n(X)$

- Par construction, P_i possède $\Theta(\alpha_i)$.
- Θ est de degré $\leqslant t$.
- $\Theta(0) = \sum_{i=1}^{n} \boldsymbol{\lambda}_i H_i(0) = \sum_{i=1}^{n} \lambda_i (Q_a \times Q_b)(\alpha_i) = (Q_a \times Q_b)(0) = a \times b.$

Sécurité : Chaque H_i est uniformément aléatoire. Vérifiez que Θ est alors aussi uniformément aléatoire parmi les polynômes de degré $\leqslant t$.

Le Protocole BGW - Étape III : Reconstruction



À la fin du protocole, chaque partie peut alors broadcaster sa $(t+1, n)$-part de $f(x_1, \ldots, x_n)$ et peut reconstruire le résultat.

Le Protocole BGW en Résumé

Le protocole de Ben-Or, Goldwasser et Wigderson (1988)

Pour n'importe quelle fonction $f(x_1, \ldots, x_n)$, il existe un protocole de MPC à n joueurs, capable de calculer f avec **Sécurité Parfaite**, en présence d'un adversaire **semi-honnête**, et controlant jusqu'à $\mathbf{t < n/2}$ parties.

Extension : Si $t < n/3$, on peut atteindre une sécurité inconditionnelle contre des adversaires malicieux (actifs). Pour une preuve précise de ce protocole dans ces deux cas, avec des simulateurs, voir `https://eprint.iacr.org/2011/136`