

Prédictions sécurisées en utilisant le calcul multipartite sécurisé

Antony Rouve Guilhem Carlet Thibault Saccon Magali Jomier

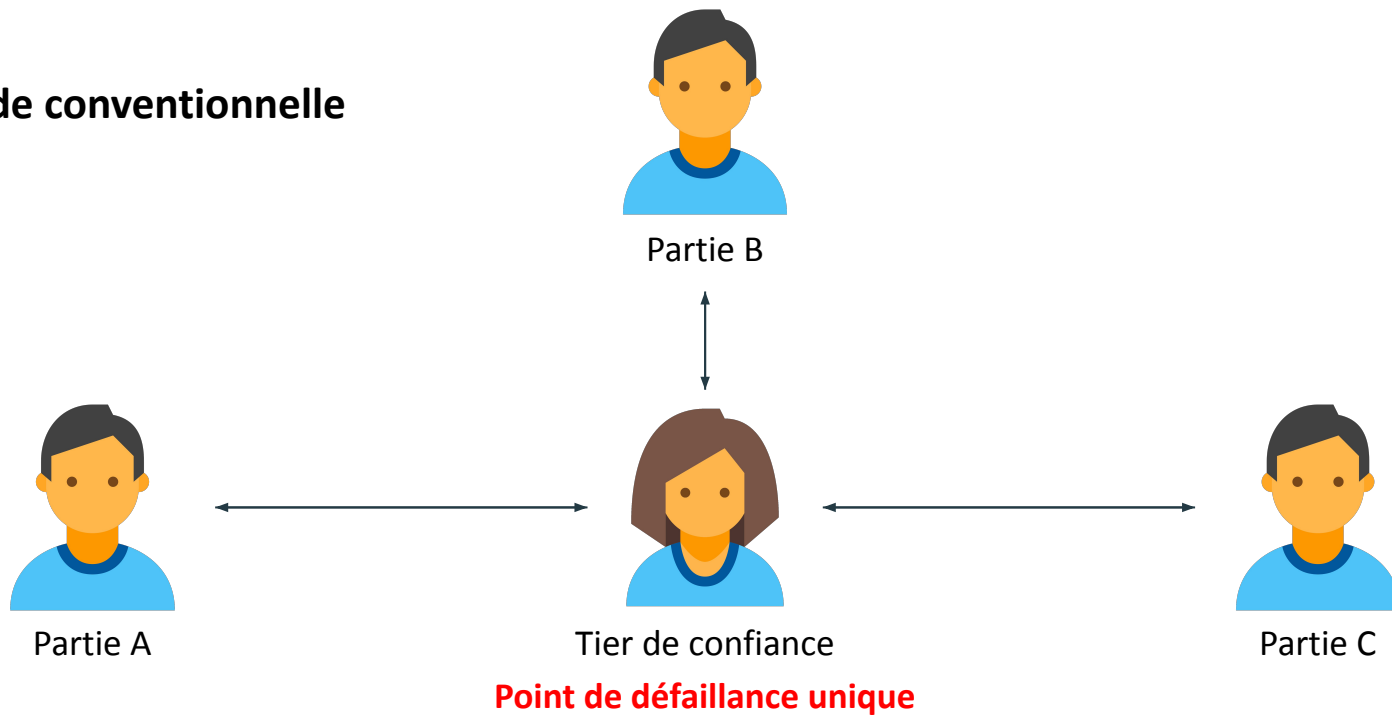


Introduction au SMC



Introduction au SMC

Méthode conventionnelle



Introduction au SMC



J'ai des données secrètes et j'ai besoin que quelqu'un les traitent sans les révéler

Est-ce seulement possible ?



Solution: **Divisez** le secret en parties que nous **reconstruons** plus tard



Partage du secret



Partage du secret

	Additive Secret Sharing	Shamir Secret Sharing	Yao's Garbled Circuit
Basé sur	Addition	L'interpolation de Lagrange	Transfert inconscient
Adapté au	Calcul arithmétique	Calcul arithmétique	Calcul booléen
Conçu pour	≥ 2 Parties	≥ 2 Parties	2 Parties

Partage du secret

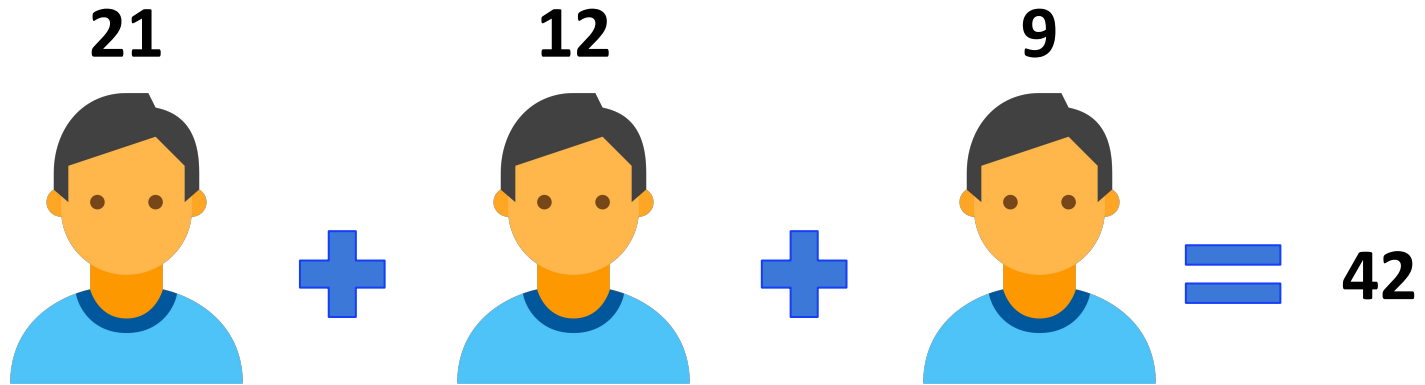
Additive Secret Sharing

- **Share**(x) : on input $x \in \mathcal{G}$, sample x_1, \dots, x_{n-1} uniformly at random from \mathcal{G} , and set $x_n \leftarrow x - \sum_{i=1}^{n-1} x_i$.
- **Reconstruct**(x_1, \dots, x_n) : output $x \leftarrow \sum_{i=1}^n x_i$.

Partage du secret

Additive Secret Sharing

Secret = 42



Partage du secret

Additive Secret Sharing

Secret = EPITA

_PI_A



E_

E_IT_



_P_T_

EP_TA



_I_A

2 Parties

3 Parties


Nombre d'acteur

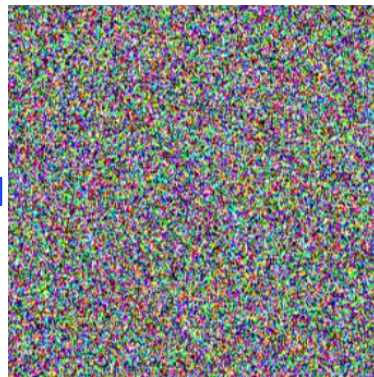
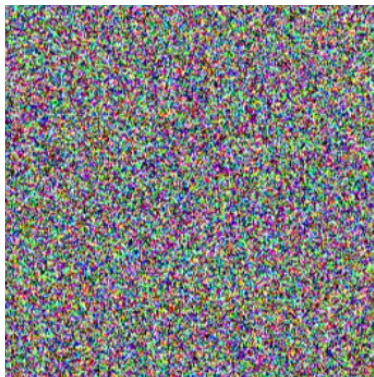
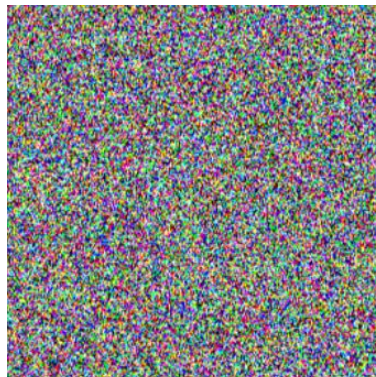
\neq

Nombre de parties requises
pour reconstruire le secret

Partage du secret

Additive Secret Sharing

Secret = 



Partage du secret

Additive Secret Sharing



- Il est parfois nécessaire de réunir toute les parties afin de reconstruire le secret
- Une part du secret révèle beaucoup trop d'information sur le secret
- Comment diviser le secret pour que 2 de toutes les parties partageant le secret soient suffisantes pour le reconstruire ?

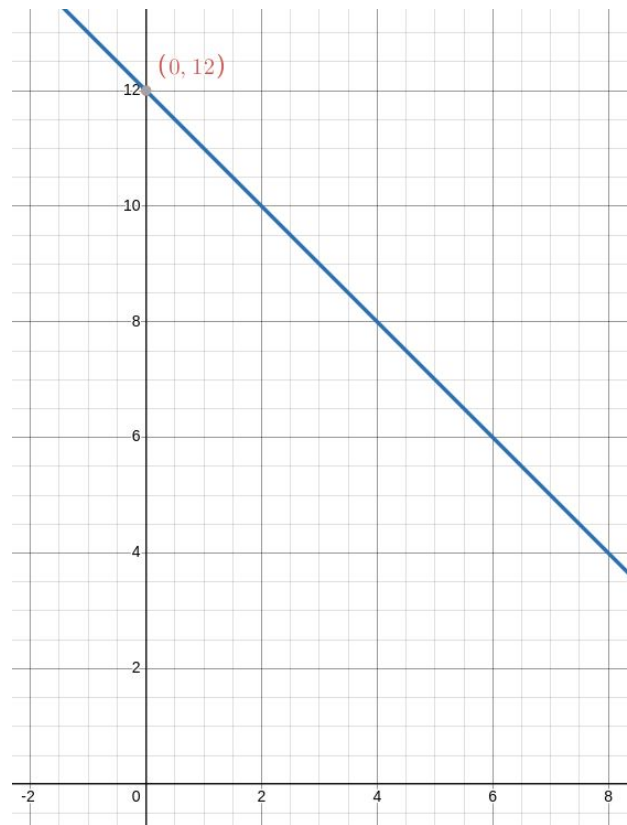
Partage du secret

Shamir Secret Sharing

Secret = **12**

Parties requises pour trouver le secret = **2**

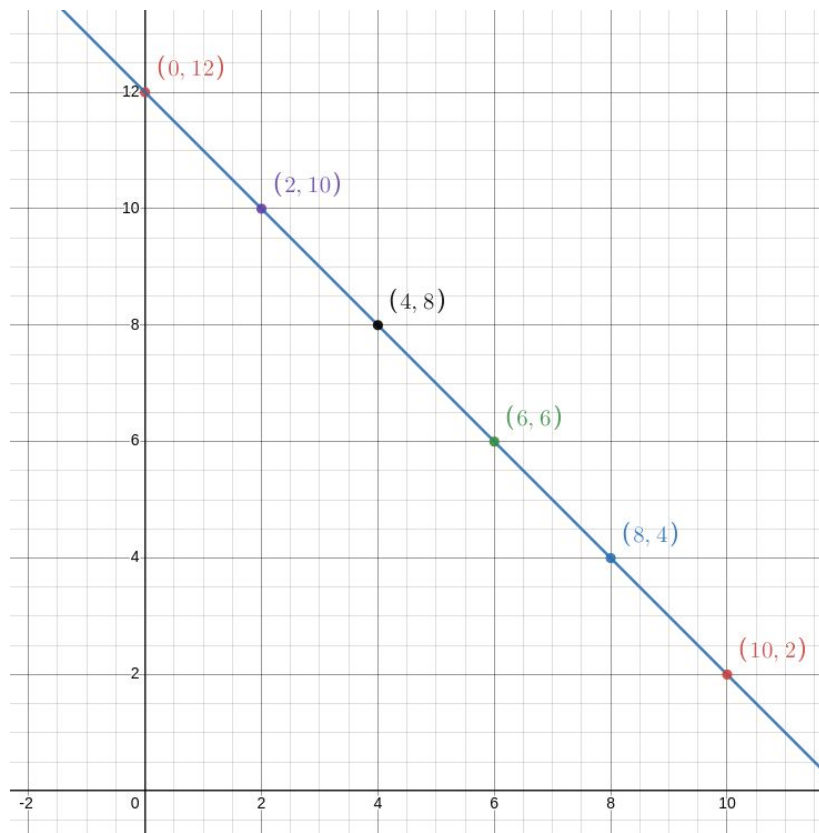
- Point secret S (**0, 12**)
- Droite secrète passant par S (**$y = -x + 12$**)



Partage du secret

Shamir Secret Sharing

- On peut choisir autant de points sur la droite que l'on veut (part du secret à distribuer)
- 2 parties suffisent pour reconstruire la ligne et trouver le secret

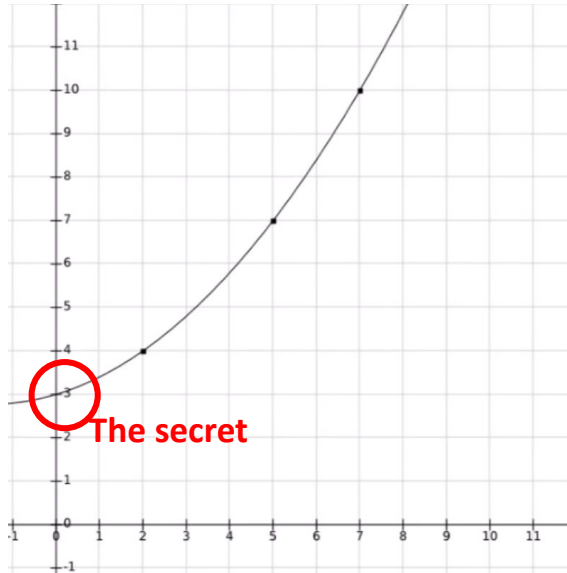
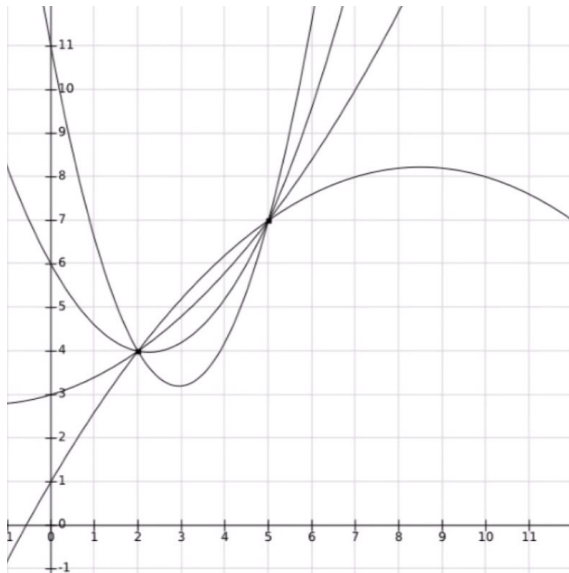
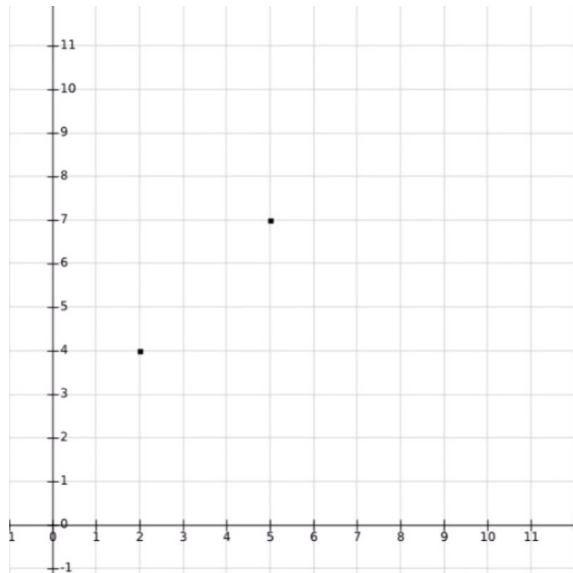


Partage du secret

Shamir Secret Sharing

Que faire si nous voulons au minimum 3 parts ?

On utilise une **parabole**



Partage du secret

Shamir Secret Sharing

Degré **1** => **2** points

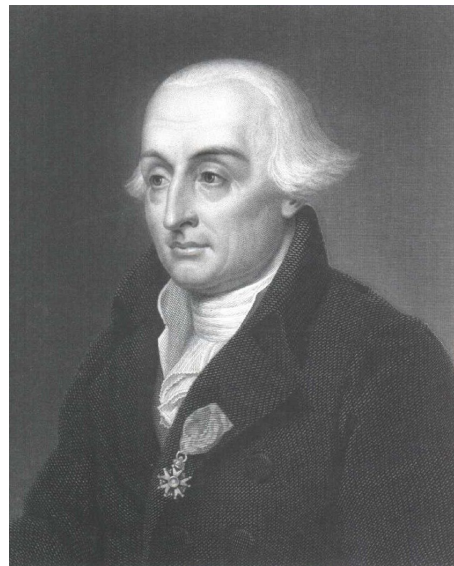
Degré **2** => **3** points

Degré **3** => **4** points

...

L'interpolation de Lagrange

Pour reconstruire un polynôme de degré n , il est nécessaire de connaître $n + 1$ points de ce polynôme



Joseph-Louis Lagrange
1736 - 1813



Les protocoles du SMC



Les protocoles du SMC

Semi-Honnête

(Adversaires passifs)

Toutes les parties suivent les spécifications du protocole mais essaient d'apprendre plus que ce qui est autorisé par ce protocole.

Malicieux

(Adversaires actifs)

Des acteurs peuvent collaborer de n'importe quelle manière et sont libres de ne pas suivre le protocole afin d'apprendre le plus d'informations possible.

Les protocoles du SMC

Les protocoles principaux du SMC

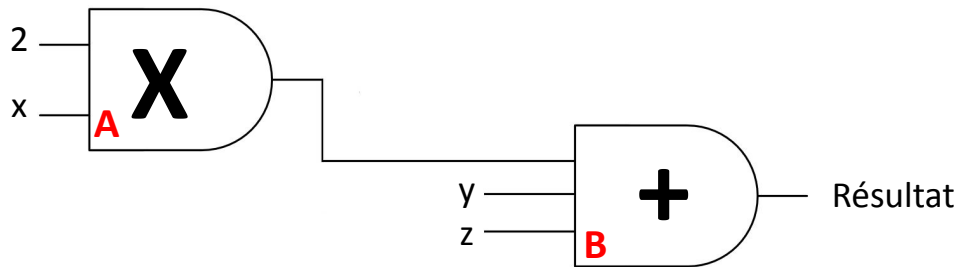
- Goldreich Micali Wigderson (GMW)
- Yao's Garbled Circuit based protocols
- Shamir Secret Sharing based protocols
- Ben-Or Goldwasser Widgerson (BGW)
- Secure Three-Party Computation (3PC)
- SPDZ protocol (Additive secret sharing)
- Tiny-OT protocol (2 Party Protocol)

Les protocoles du SMC

Goldreich Micali Wigderson, GMW Protocol

Étape 1: Écrire la fonction public comme un circuit

$$f(x, y, z) = 2x + y + z$$



Les protocoles du SMC

Goldreich Micali Wigderson, GMW Protocol

Étape 2: Partager nos secrets



Secret $x \Rightarrow x_1 + x_2 + x_3$

Secret $y \Rightarrow y_1 + y_2 + y_3$

Secret $z \Rightarrow z_1 + z_2 + z_3$

Ici par exemple, nous avons 3 secrets que l'on partage en 3 parts pour nos 3 parties.

Les protocoles du SMC

Goldreich Micali Wigderson, GMW Protocol

Étape 3: Distribution des secrets



Secret $x \Rightarrow x_1 + x_2 + x_3$

Secret $y \Rightarrow y_1 + y_2 + y_3$

Secret $z \Rightarrow z_1 + z_2 + z_3$



x_1, y_1, z_1



x_2, y_2, z_2



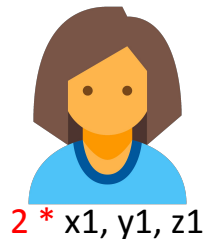
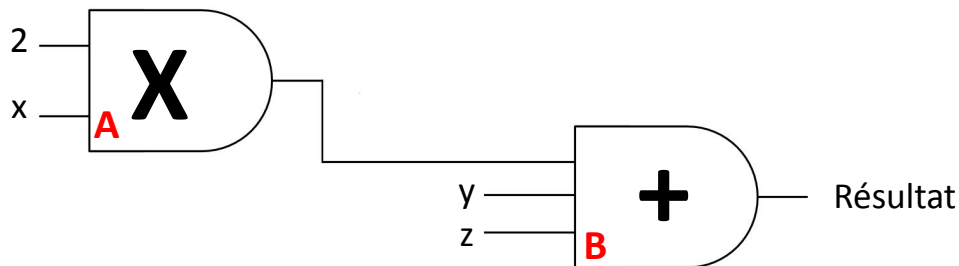
x_3, y_3, z_3

Les protocoles du SMC

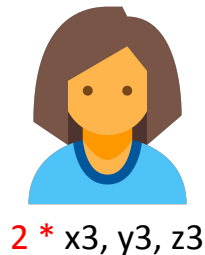
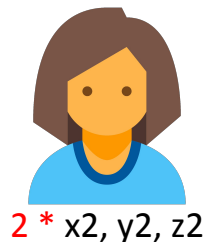
Goldreich Micali Wigderson, GMW Protocol

Étape 4: Évaluer le circuit public porte par porte

Porte A



Pas besoin de
communiquer

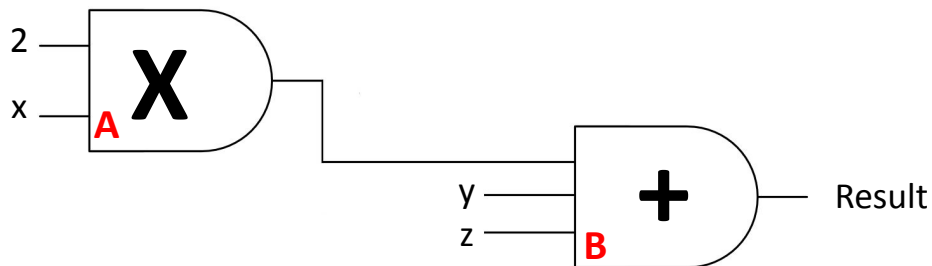



Les protocoles du SMC

Goldreich Micali Wigderson, GMW Protocol


Étape 4: Évaluer le circuit public porte par porte


Porte B




$$2 * x1 + y1 + z1$$

Pas besoin de
communiquer


$$2 * x2 + y2 + z2$$

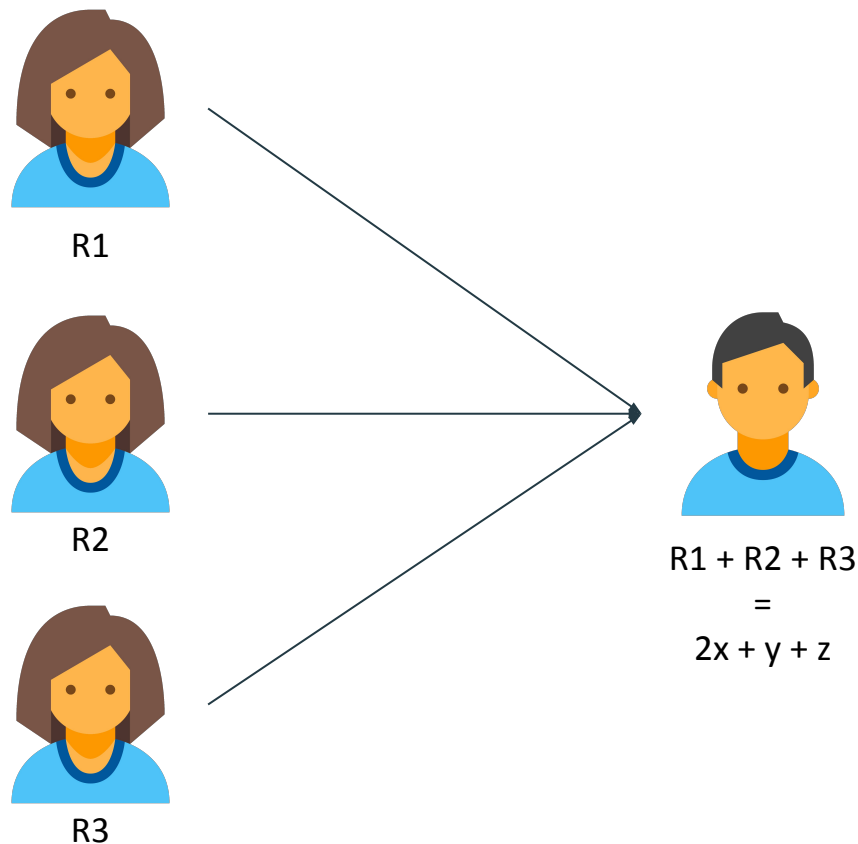

$$2 * x3 + y3 + z3$$

Les protocoles du SMC

Goldreich Micali Wigderson, GMW Protocol

Étape 5: Réassembler les sorties

$$f(x, y, z) = 2x + y + z$$





Le problème



Le problème



Le problème

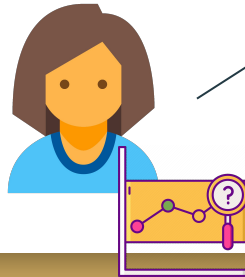
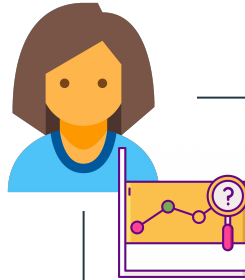
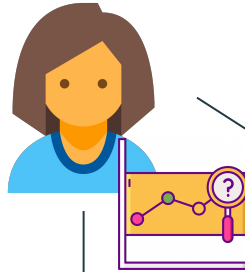
L'architecture choisie

Partie d'entrée



Données
secrètes

Parties de calcul



Partie de sortie

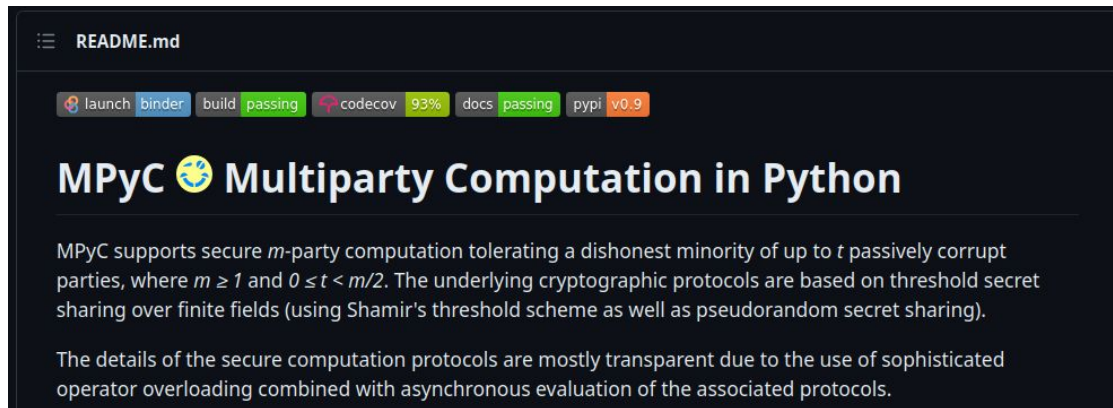


Prédiction

Le problème



<https://github.com/lschoe/mpyc>

A screenshot of the MPyC GitHub repository README. At the top, it says 'README.md'. Below that are several status badges: 'launch', 'binder', 'build passing', 'codecov 93%', 'docs passing', and 'pypi v0.9'. The title 'MPyC' is followed by a globe icon and the text 'Multiparty Computation in Python'. The main text describes that MPyC supports secure m -party computation tolerating a dishonest minority of up to t passively corrupt parties, where $m \geq 1$ and $0 \leq t < m/2$. It mentions that the underlying cryptographic protocols are based on threshold secret sharing over finite fields, using Shamir's threshold scheme as well as pseudorandom secret sharing. The details of the secure computation protocols are mostly transparent due to the use of sophisticated operator overloading combined with asynchronous evaluation of the associated protocols.

launch binder build passing codecov 93% docs passing pypi v0.9

MPyC 🌐 Multiparty Computation in Python

MPyC supports secure m -party computation tolerating a dishonest minority of up to t passively corrupt parties, where $m \geq 1$ and $0 \leq t < m/2$. The underlying cryptographic protocols are based on threshold secret sharing over finite fields (using Shamir's threshold scheme as well as pseudorandom secret sharing).

The details of the secure computation protocols are mostly transparent due to the use of sophisticated operator overloading combined with asynchronous evaluation of the associated protocols.

MPyC is a Python package for secure multiparty computation (MPC).

MPyC provides a runtime for performing computations on secret-shared values, where parties interact by exchanging messages via peer-to-peer connections.

The MPC protocols are based on Shamir's threshold secret sharing scheme and withstand passive adversaries controlling less than half of the parties.

Le problème

Un exemple:

```
from mpyc.runtime import mpc

secint = mpc.SecInt()

mpc.run(mpc.start())

x_share = mpc.input(secint(42), senders=0)
y_share = mpc.input(secint(12), senders=0)
z_share = mpc.input(secint(36), senders=0)

@mpc.coroutine
async def function(x_share, y_share, z_share):
    return x_share + y_share + z_share

result = mpc.run(mpc.output(function(x_share, y_share, z_share)))

print("Result of the computation: ", result)

mpc.run(mpc.shutdown())
```

Initialise toutes les parties

Partage de façon secrète les nombres 42,12 et 36

Fonction qui va s'exécuter sur l'ensemble des parties

Exécuter la fonction avec les paramètres donnés

Déconnecter toutes les parties

Le problème

Quelques statistiques



Total: **1272**

Pas de maladie: **1106**

Maladie coronarienne: **166**

87 % Taux de bonnes prédiction

99 % Taux de bonnes prédiction sur la population ne développant pas de maladie coronarienne après 10 ans

7 % Taux de bonnes prédictions sur la population développant une maladie coronarienne après 10 ans

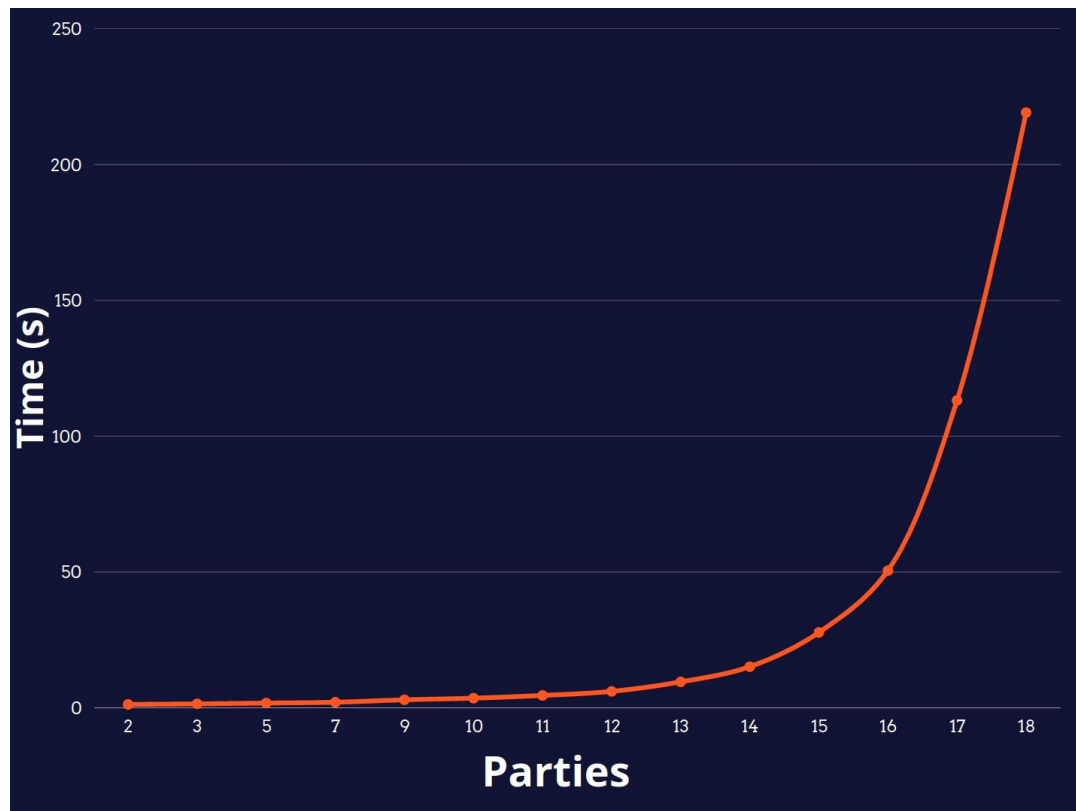
Le problème

Quelques statistiques

Spécifications de l'ordinateur
de test :

CPU: Ryzen 7 5700X

RAM: 25Go





Conclusion



Conclusion



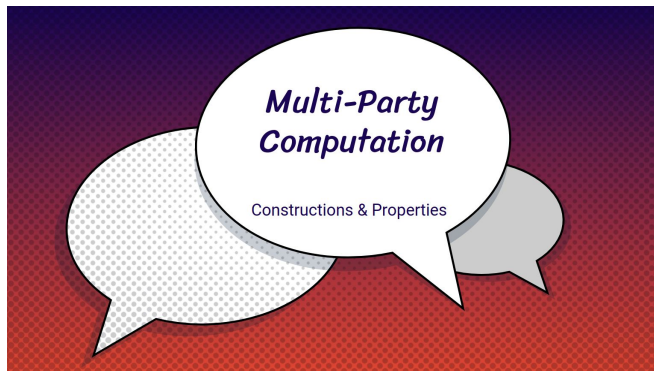
- Assure la confidentialité des données et du modèle
- Résistant contre des adversaires puissants en termes de calcul
- Peut être sécurisé contre les parties malicieuses
- Peut être appliqué à d'autres domaines



- Lent à cause des échanges
- Demande une grande puissance de calcul
- Complexe à mettre en place à grande échelle

Sources

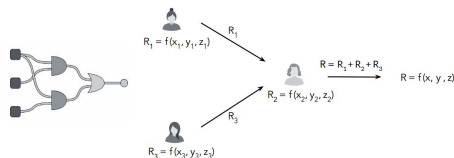
<https://www.di.ens.fr/~nitulesc/files/slides/MPC-intro.pdf>



MPC from Secret Sharing

The GMW/BGW* Approach:

- The (public) function being computed is written as a circuit
- Each participant secret-shares their private input
- The circuit is evaluated gate-by-gate on the shares (this requires communication between participants)
- Answer is reconstructed from final shares

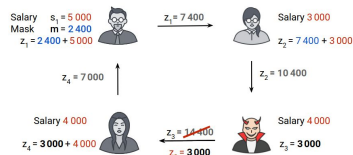


29

MPC - Security

We need to make sure that an **adversary** cannot learn the private data of the honest parties.

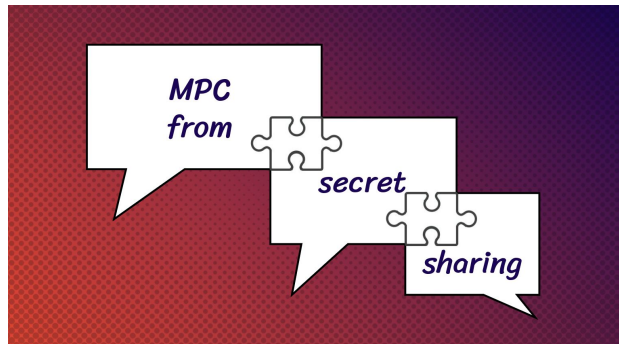
We assume that **communication channels** are private and authenticated.



Security Model: The adversary is inside the system and corrupts some of the parties

- **Semi-honest (passive)** – Corrupted players follow the protocol but try to learn more → private computation
- **Malicious (active)** – Corrupted players can collaborate in any way and misbehave arbitrarily → secure computation

12



Sources

https://en.wikipedia.org/wiki/Garbled_circuit

https://en.wikipedia.org/wiki/Shamir%27s_secret_sharing

https://www.youtube.com/watch?v=kkMps3X_tEE

<https://www.youtube.com/watch?v=iFY5SyY3IMQ>

<https://www.youtube.com/watch?v=xwxkp4fMWsk>

<https://www.youtube.com/watch?v=K54ildEW9-Q>

https://sands.edpsciences.org/articles/sands/full_html/2022/01/sands20210001/sands20210001.html

<https://wiki.mpcalliance.org/protocols.html>

<https://www.youtube.com/watch?v=HOqv5xzrIFl>

<https://eprint.iacr.org/2019/1390.pdf>

<https://www.cs.purdue.edu/homes/hmaji/teaching/Fall%202017/lectures/39.pdf>

<https://eprint.iacr.org/2020/1577.pdf>

https://link.springer.com/chapter/10.1007/978-3-030-17659-4_15



Merci de votre
attention

