E-voting based on Homomorphic Encryption

Crypto projects 2023
EPITA – SRS – CRYPI

Adrien Langou - Aurélien Rebourg
Pierre Blaess - Lucas Theze

01 Introduction

Définition du chiffrement homomorphe

02 Objectifs

Enjeux liés au *e-voting*

03 Recherche

Choix des bibliothèques utilisées pour l'implémentation

04 Protocole

Présentation de la mise en place du protocole

05 Implémentation

Implémentation du système de e-voting

06 Conclusion

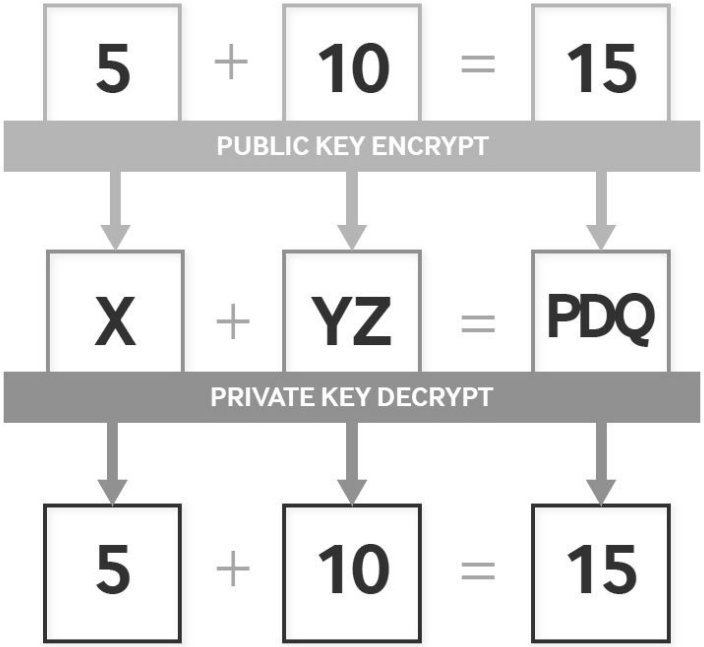
Prise de recul sur les résultats

01

Introduction

Le chiffrement homomorphe

Introduction



Une forme de chiffrement permettant d'effectuer des calculs sur des données chiffrées

Introduction

3 types de chiffrements homomorphes

- Le chiffrement homomorphe **partiel**
- Le chiffrement **quelque peu** homomorphe
- Le chiffrement homomorphe **complet**

- **UNE** opération (+ OU *) et calculs **illimités**
- **PLUSIEURS** opérations mais calculs **limités**
- **PLUSIEURS** opérations et calculs **illimités**

Introduction

3 types de chiffrements homomorphes

- Le chiffrement homomorphe **partiel**	- **UNE** opération (+ OU *) et calculs **illimités**
- Le chiffrement **quelque peu** homomorphe	- **PLUSIEURS** opérations mais calculs **limités**
- Le chiffrement homomorphe **complet**	- **PLUSIEURS** opérations et calculs **illimités**

02

Objectifs

Les enjeux liés au e-voting

Objectifs

Les votes doivent être anonymes

Objectifs

Les votes ne doivent
pas être modifiés

03

Recherches

Choix des bibliothèques utilisées
pour l'implémentation

Recherches

Matérialisation d'un vote :

[Jean, Michel, Alice, Bob]

Vote pour **Alice** :

[0, 0, 1, 0]

Recherches

[Jean, Michel, Alice, Bob]

Plusieurs votes :

[0, 0, 1, 0]

[0, 1, 0, 0]

[0, 0, 1, 0]

Résultat :

[0, 1, 2, 0]

Alice a gagné !

Recherches

Pyfhel

Pyfhel

Pyfhel implémente les fonctionnalités de plusieurs opérations de chiffrement homomorphe dont **l'addition**

Numpy

NumPy est une bibliothèque ajoutant la prise en charge de grands **tableaux et matrices** multidimensionnels

04

Protocole

Présentation de la mise en place du protocole

Protocole

Communication avec 2 serveurs :

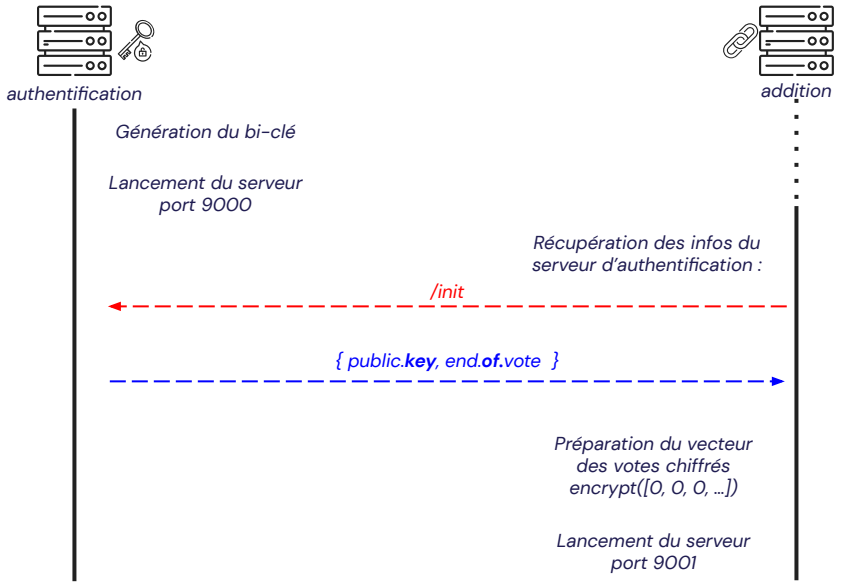
- **Serveur d'authentification**
  ➔ Éligibilité et authentification
  ➔ Bi-clés de chiffrement
- **Serveur d'addition**
  ➔ Additions des votes
  ➔ Résultat des votes (chiffrés)

Protocole

Énonciation du protocole :

1. **<u>Initialisation des serveurs :</u>**

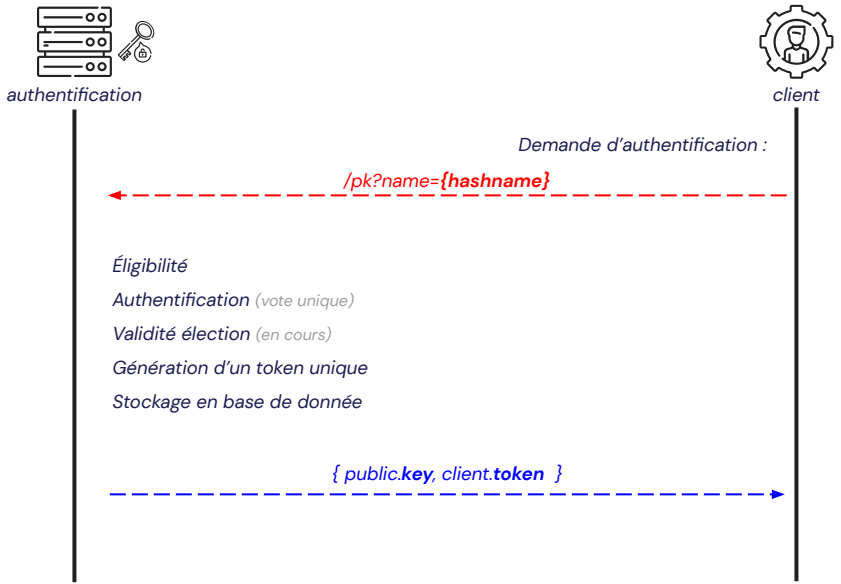
Le serveur d'authentification se lance en premier



Protocole

Énonciation du protocole :

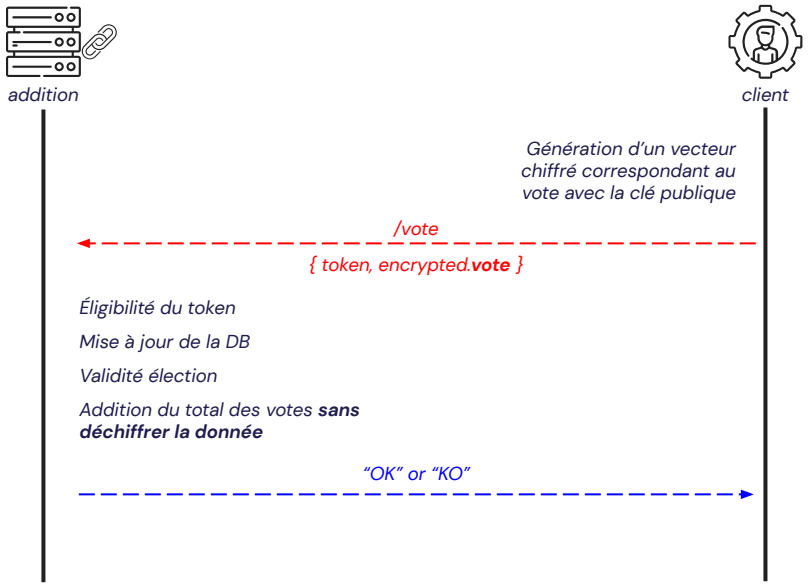
2. Authentification du client



Protocole

Énonciation du protocole :

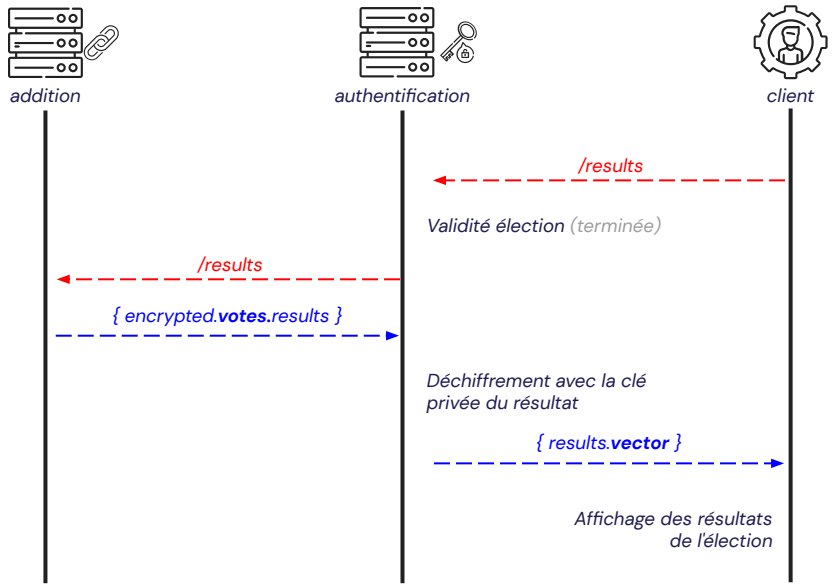
3. Vote du client



Protocole

Énonciation du protocole :

4. **Résultats des votes**



05

Implémentation

Implémentation du système de e-voting

Implémentation

Python

➔ Facilité d'utilisation
➔ Nombreuses bibliothèques

Requêtes HTTP

→ **Réception** et **traitement** en utilisant des serveurs **Flask.**

→ Communication entre les serveurs par l'**émission** de requêtes avec des paramètres **JSON**.

Implémentation

Chiffrement homomorphe

Avec Pyfhel :

- Création d'un objet de chiffrement homomorphe
- Définition du context

  Paramètres de sécurité et de performance du schéma de chiffrement.
  Bi-clés, algorithmes utilisés, fonctions de calcul homomorphes.

```
if __name__ == "__main__":
    HE = Pyfhel()
    HE.contextGen(scheme='bfv', n=2**12, t_bits=20)
    HE.keyGen()
    init_data()

    app.run(host="127.0.0.1", port=9000)
```

scheme='bfv' : Schéma de chiffrement homomorphe BGV *(Brakerski-Gentry-Vaikuntanathan)* avec la variante FV *(Fan-Vercauteren)*.

n=2**12 : Taille de la clé de chiffrement utilisée *(4096 bits)*.

t_bits=20 : Nombre de bits utilisé pour stocker chaque élément du corps de chiffrement *(sur 20 bits)*.

```
ENCRYPT_RESULT += PyCtxt(pyfhel=HE, bytestring=cyphervalue)
```

Addition des votes *(serveur d'addition)*

Implémentation

Initialisation

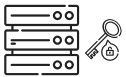
```
client@42sh$ python src/client.py
[E-Voting Client] Please enter your name :
> Bob
[E-Voting Client] Here are the candidates :
- 0 Pierre Olivier Mercier
- 1 Sebastien Bombal
- 2 Constance Beguier
- 3 Julien Sterckeman
[E-Voting Client] Please enter a candidate key :
> 0
[E-Voting Client] Your vote has successfully been completed.
client@42sh$
```

```
client@42sh$ python server_auth.py
usage: CRIPY [-h] --endtime ENDTIME
CRIPY: error: the following arguments are required: --endtime
client@42sh$ python server_auth.py --endtime 3600
 * Serving Flask app 'server_auth'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a product
ion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9000
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:50:06] "GET /init HTTP/1.1" 200 -
442e5aa74904cc73bf6fb54d1d54d333bd596aa9bb4bb4e961 HTTP/1.1" 200
-
```

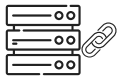
```
addition@42sh$ python server_addition.py
 * Serving Flask app 'server_addition'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a produc
tion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9001
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:50:23] "POST /vote HTTP/1.1" 200 -
```

Client

Serveur d'authentification



Serveur d'addition



Implémentation

Authentification

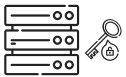
```
client@42sh$ python src/client.py
[E-Voting Client] Please enter your name :
> Bob
[E-Voting Client] Here are the candidates :
- 0 Pierre Olivier Mercier
- 1 Sebastien Bombal
- 2 Constance Beguier
- 3 Julien Sterckeman
[E-Voting Client] Please enter a candidate key :
> 0
[E-Voting Client] Your vote has successfully been completed.
client@42sh$
```

```
client@42sh$ python server_auth.py
usage: CRIPY [-h] --endtime ENDTIME
CRIPY: error: the following arguments are required: --endtime
client@42sh$ python server_auth.py --endtime 3600
 * Serving Flask app 'server_auth'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a product
ion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9000
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:50:06] "GET /init HTTP/1.1" 200 -
127.0.0.1 - - [10/May/2023 11:50:23] "GET /pk?name=cd9fb1e148ccd8
442e5aa74904cc73bf6fb54d1d54d333bd596aa9bb4bb4e961 HTTP/1.1" 200
```

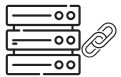
```
addition@42sh$ python server_addition.py
 * Serving Flask app 'server_addition'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a produc
tion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9001
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:50:23] "POST /vote HTTP/1.1" 200 -
```

Client

Serveur d'authentification



Serveur d'addition



Implémentation

Vote

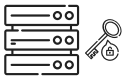
```
client@42sh$ python src/client.py
[E-Voting Client] Please enter your name :
> Bob
[E-Voting Client] Here are the candidates :
- 0 Pierre Olivier Mercier
- 1 Sebastien Bombal
- 2 Constance Beguier
- 3 Julien Sterckeman
[E-Voting Client] Please enter a candidate key :
> 0
[E-Voting Client] Your vote has successfully been completed.
client@42sh$
```

```
client@42sh$ python server_auth.py
usage: CRIPY [-h] --endtime ENDTIME
CRIPY: error: the following arguments are required: --endtime
client@42sh$ python server_auth.py --endtime 3600
 * Serving Flask app 'server_auth'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a product
ion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9000
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:50:06] "GET /init HTTP/1.1" 200 -
127.0.0.1 - - [10/May/2023 11:50:23] "GET /pk?name=cd9fb1e148ccd8
442e5aa74904cc73bf6fb54d1d54d333bd596aa9bb4bb4e961 HTTP/1.1" 200
-
```

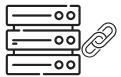
```
addition@42sh$ python server_addition.py
 * Serving Flask app 'server_addition'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a produc
tion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9001
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:50:23] "POST /vote HTTP/1.1" 200 -
```

Client

Serveur d'authentification



Serveur d'addition



Implémentation

Résultats (cas d'erreur)

```
client@42sh$ python src/client.py
[E-Voting Client] Please enter your name :
> Bob
[E-Voting Client] Here are the candidates :
- 0 Pierre Olivier Mercier
- 1 Sebastien Bombal
- 2 Constance Beguier
- 3 Julien Sterckeman
[E-Voting Client] Please enter a candidate key :
> 0
[E-Voting Client] Your vote has successfully been completed.
client@42sh$ python src/client.py -R
[E-Voting Client] Asking the server for vote results
[E-Voting Client] Error the vote is not finished
```

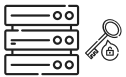
```
auth@42sh$ python server_auth.py --endtime 30
 * Serving Flask app 'server_auth'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a product
ion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9000
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:55:13] "GET /init HTTP/1.1" 200 -
127.0.0.1 - - [10/May/2023 11:55:19] "GET /pk?name=cd9fb1e148ccd8
442e5aa74904cc73bf6fb54d1d54d333bd596aa9bb4bb4e961 HTTP/1.1" 200

127.0.0.1 - - [10/May/2023 11:55:24] "GET /results HTTP/1.1" 200
```

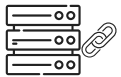
```
addition@42sh$ python server_addition.py
 * Serving Flask app 'server_addition'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a produc
tion deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:9001
Press CTRL+C to quit
127.0.0.1 - - [10/May/2023 11:55:19] "POST /vote HTTP/1.1" 200 -
```

Client

Serveur d'authentification

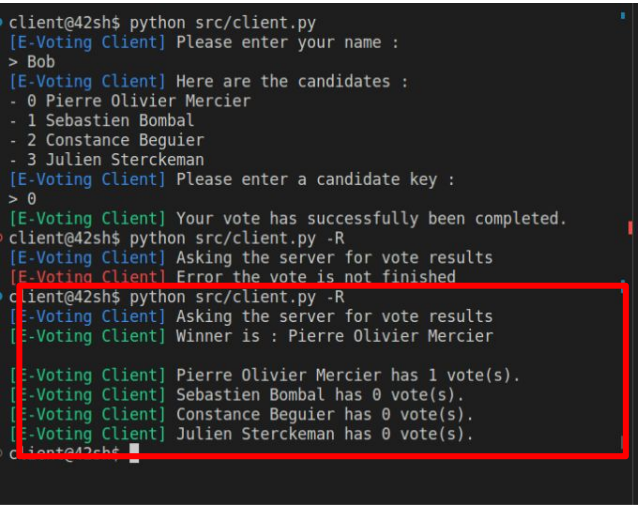


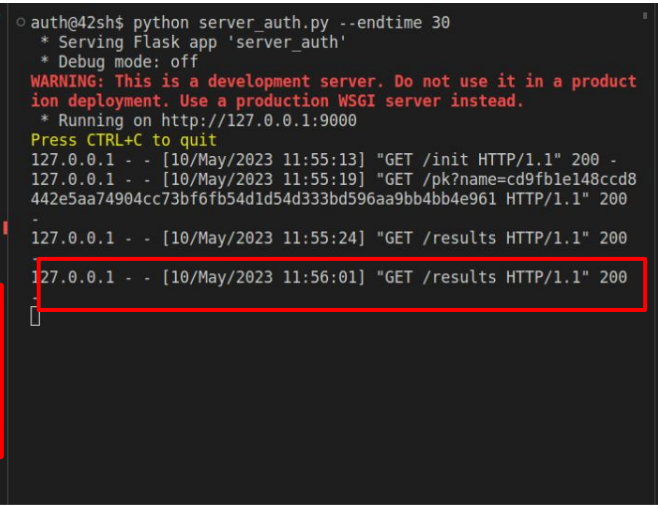
Serveur d'addition

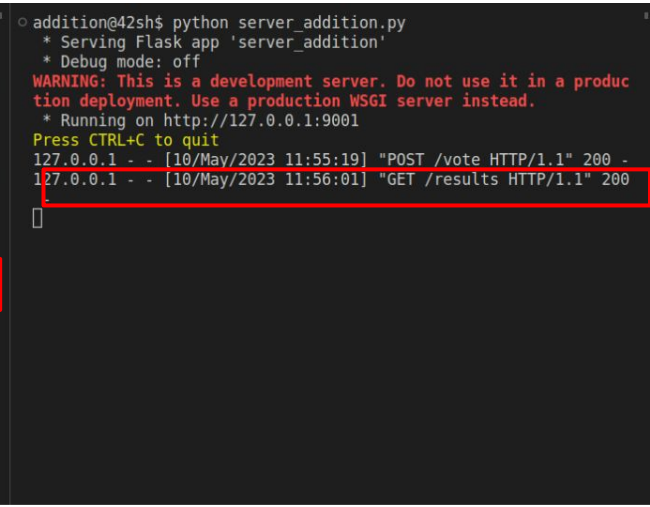


Implémentation

Résultats

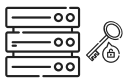




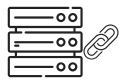


Client

Serveur d'authentification



Serveur d'addition



Implémentation

Base de donnée

db.json > 776d7da8c46355d226720fadd34de83d

```
{
    "8aa793a7c5d5ce0be43fc3a1c5258f3c": true,
    "8c54543aa58c8980b3354d7244f5d9f3": true,
    "01e084559705c5fab43ef3c4057bd8fb": true,
    "b53fe4ecf056492855d193a3136c225e": true,
    "776d7da8c46355d226720fadd34de83d": true,
    "00d6519915d70ce3118969f72906eeeb": true,
    "f80e4836843ecb85ef442170fa4d0ab5": true,
    "0b8b870a4e4fa42c94353048655462e7": true,
    "4c7f7ac5711404d9340c0b5af8921646": false,
    "71ad8a0a11b17fc3d920c16c44b1158b": false,
    "3b39c88699265410ad95f47e6f6dbaf9": true,
    "1bf05c7d2945a1c0030dd9886ef9ea42": true,
    "f79f56bc43419c238b68ce44cf936e48": true,
    "b9886fbf4f547b3335fc84ab85b302e2": true,
    "0042617c3ccc5a807ee192b161b7c44b": true,
    "c586d3a620e05eb07b48570015eed8fa": true,
    "e3f9083473fc0788d4eeb4f5397f698d": true,
    "32800bac821ed4737aa11b3df293621a": true,
    "2c99ce3c4154d553f4d6f5a83e61753e": true,
    "232467b492fbda57d591bf491a24b882": true,
    "2caa8b27df0773cf85b5e85d065dcc66": true,
    "20ccaad02711e4af999be2435901df78": true,
    "04b3b4693acd59c8fcaeb7a9626eadb5": true,
    "716ef6b2720ce4d8241a2889cc2c0dc5": true,
    "551df380b263c6ce11d05f8cb80e846e": true,
    "77b480b23df924ee57b2cf140f89ee59": false,
    "8fb75596b7a6d2289f3e1e3939a7f904": true,
    "a974b5407ac308700b5ce9a8ad833445": false,
    "2abd849d838fc6d5569fb2b1042f59d7": true,
    "cd8f2bb030810cf3b6b900fddaed6a37": false,
    "bdab1c7c846338731a043ca9bd3c8ffd": false,
    "fbc780fd87aba59354d0a732e62a2e7a": true,
```

Format JSON

Commune aux deux serveurs

Stockage des tokens avec leur validité ou non

06

Conclusion

Prise de recul sur les résultats

Conclusion

- Algorithme coûteux en **performance** et en **vitesse**
- Bien réfléchir à son utilisation (implémentation de 2 serveurs)
- Ne pas se reposer sur la sécurité de l'algorithme de chiffrement

Merci !

Avez-vous des questions ?