ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

le cnam

Présentation du projet de fin d'étude

CLUSTERING AVEC DES DONNÉES MANQUANTES DANS LE CONTEXTE DE CHIFFREMENT HOMOMORPHE

Réalisé par :

M. AZIZ Rezak

Encadrée par :

Mme. BOUZEFRANE Samia (CNAM)

M. AUDIGIER Vincent (CNAM)

Mme. BOUZAR Lydia (ESI)

2021 - 2022

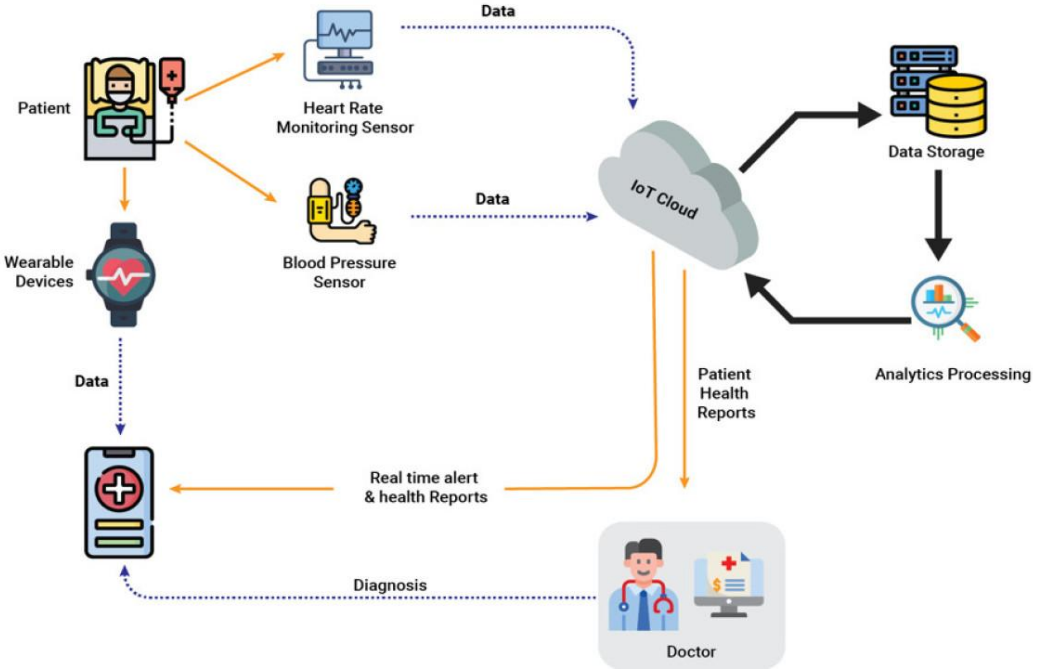
Plan de la présentation

- **Introduction**
- **État de l'art**
  - **Clustering et chiffrement homomorphe**
  - **Clustering avec des données manquantes**
- **K-means et TFHE**
- **Conception**
  - **Architecture kmeans-HE**
  - **Explication détaillée**
  - **Prise en compte données manquantes**
  - **Outils utilisés**
- **Résultats obtenus**
- **Conclusion**

CONTEXTE ET OBJECTIFS

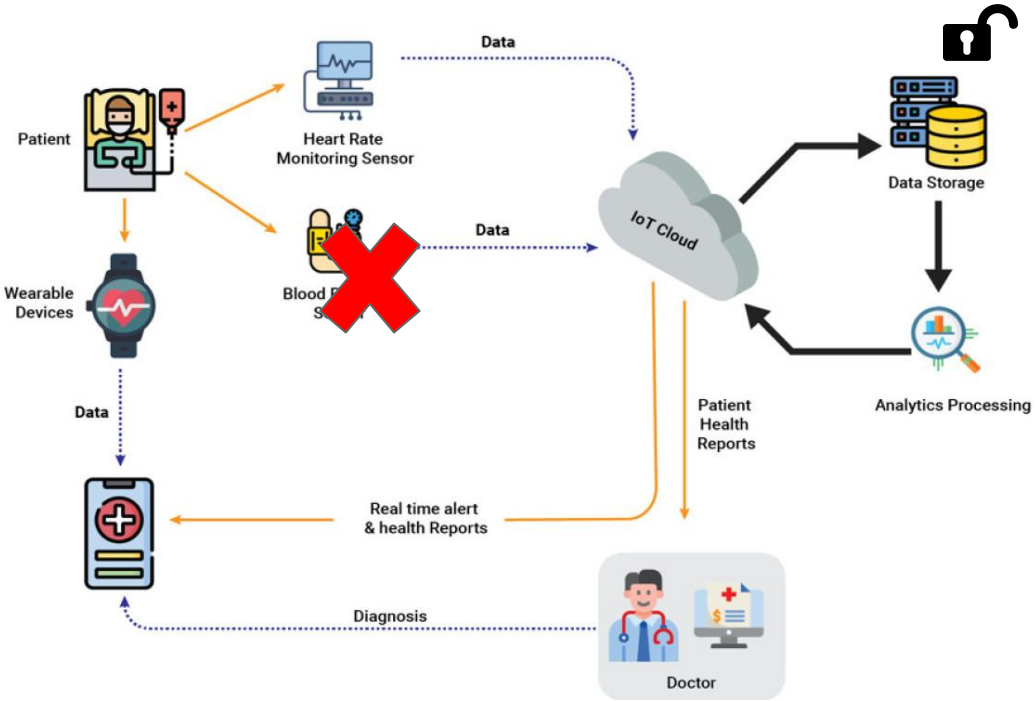
Contexte

- **Collection de large quantités de données.**
- **Utilisation des méthodes de clustering pour analyser ces données.**
- **Apparition d'un nouveau paradigme en cloud computing : MLaaS.**



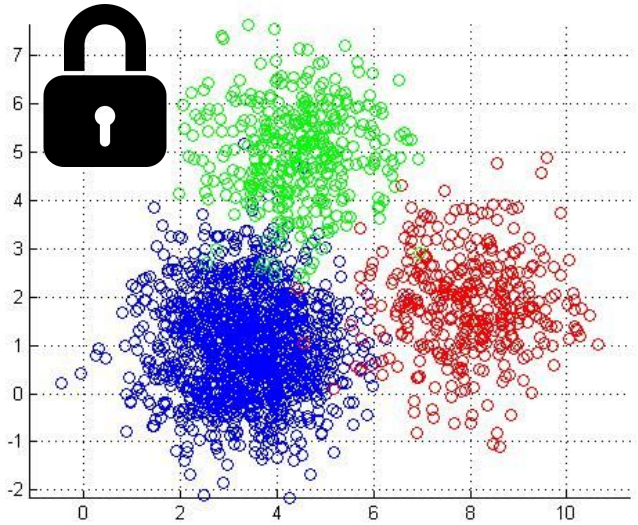
Problématiques

- **Traitement des données sensibles en clair sur le cloud. ⇒ Problème de vie privée.**
- **Défaillance des mécanismes de collecte des données. ⇒ Données incomplètes.**



Objectifs

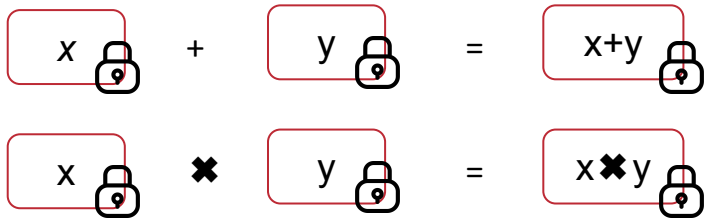
- **Effectuer un clustering de façon sécurisée sur le cloud grâce au chiffrement homomorphe.**
- **Prendre en compte la réalité sur les données manquantes.**
- **Limiter les calculs au niveau de propriétaire des données.**



ÉTAT DE L'ART

Chiffrement complètement homomorphe

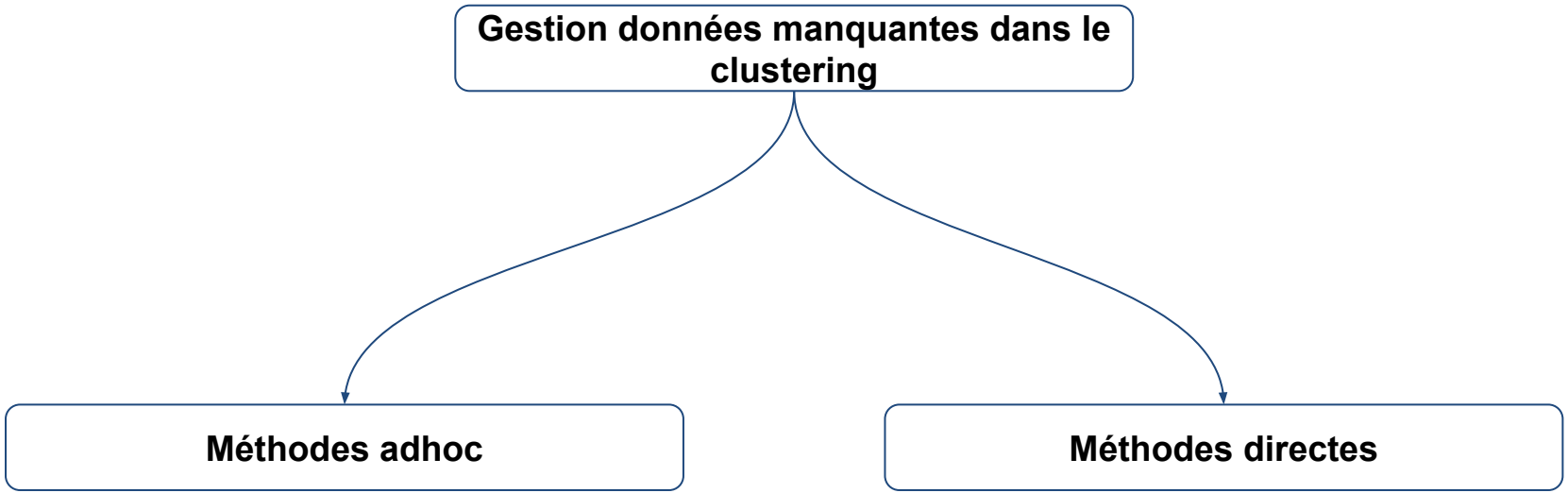
- **Un chiffrement qui permet de faire des calculs sur des données chiffrées sans les déchiffrer.**
- **Les résultats des calculs sont aussi chiffrés.**
- **Plusieurs schémas de chiffrement : (BGV ,2011), (CKKS, 2016), (TFHE, 2017),...**
- **Les seules opérations possibles sont l'addition et la multiplication.**



Clustering et chiffrement homomorphe

- **Le clustering nécessite des opérations plus complexes : comparaison et division.**
- **Solutions existantes utilisent des déchiffrements intermédiaires.**
- **Limites des solutions existantes**
    - **Trop d'opérations au niveau de propriétaire de données.**
    - **Nécessité de faire confiance à une tierce partie.**

Clustering et données manquantes



- Analyse de cas complet
- Imputation simple

- kPODs (chi et al., 2016)
- Ignorable GMM (Marbac et al., 2019)

PROPOSITION

Proposition

- **Étudier k-means (Forgy, E.W. (1965)).**
- **Proposer une solution k-means en utilisant le schéma TFHE (chillotti et al.,2016).**
- **Augmenter la solution proposée pour prendre en compte les données manquantes en utilisant la méthode des kPODs (chi et al., 2016).**

K-MEANS

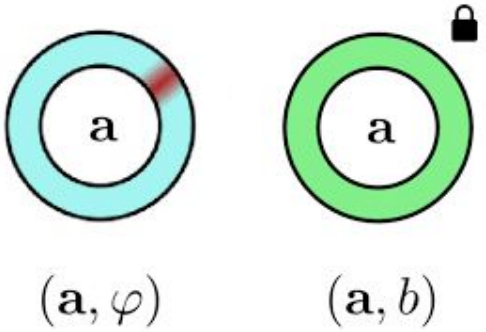
- **Minimiser la fonction :** $\sum_{k=1}^{K} \sum_{i \in C_k} d\left(Z_i, c_k\right)^2$
- **L'algorithme k-means suit l'allure suivante :**

1. Initialisation des centres initiaux aléatoirement.
2. Répéter jusqu'à convergence {
   Affectation
   Mise à jour des centres
   }

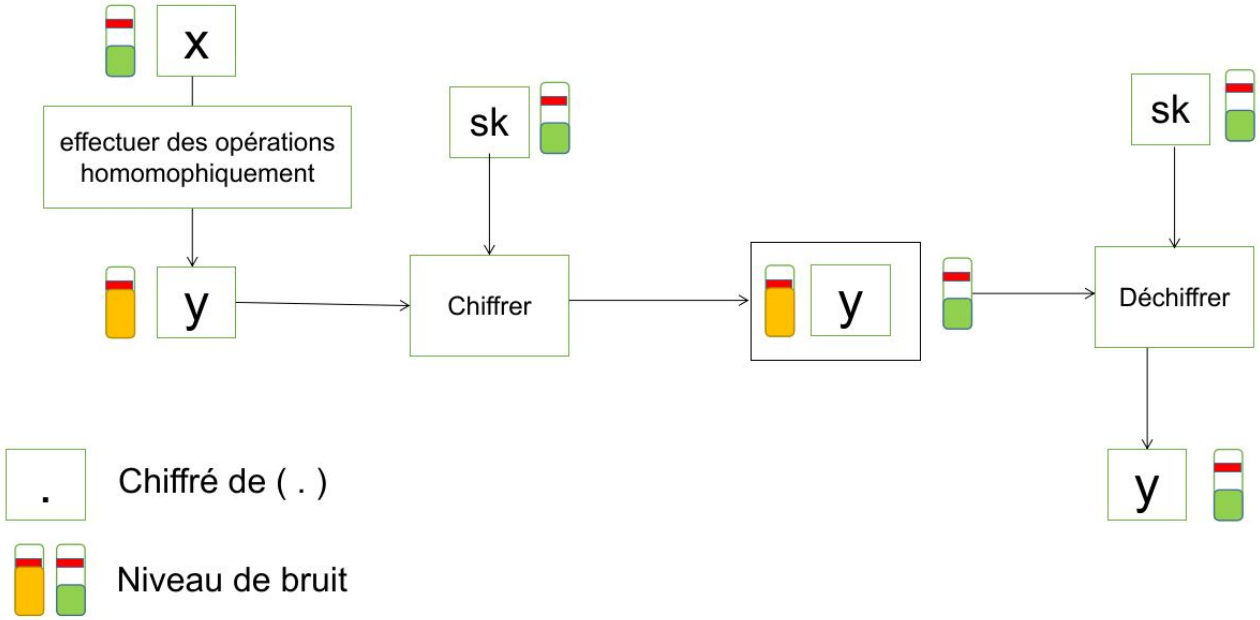
- **Opérations problématiques :**
  - **Comparaisons au niveau de l'étape d'affectation.**
  - **Division au niveau de la mise à jour des centres.**

TFHE

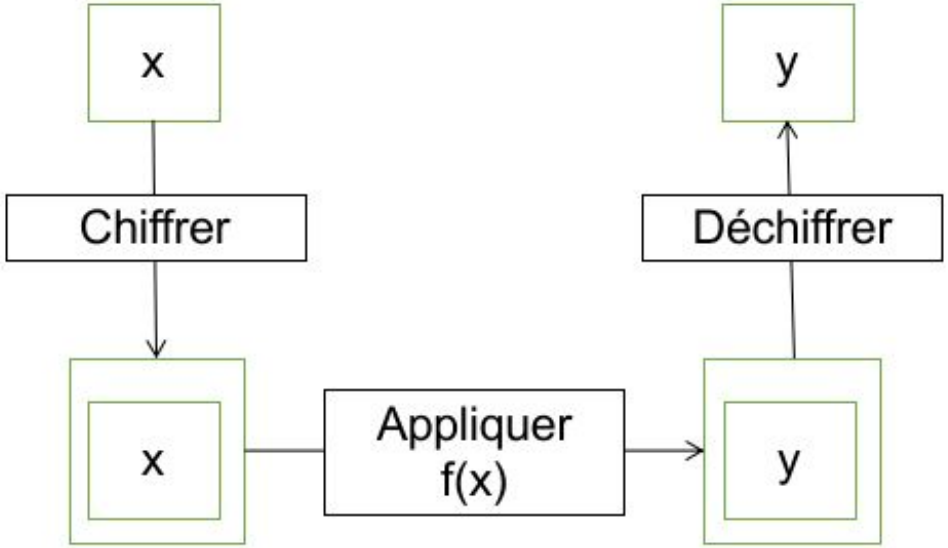
- **Un schéma basé sur le problème Learning with Error.**
- **Encode et chiffre les données sur un tore (intervalle [0,1[).**
- **Opérations possibles :**
  - **addition.**
  - **multiplication par un scalaire.**
  - **Bootstrapping programmable et fonctionnel.**



Bootstrapping



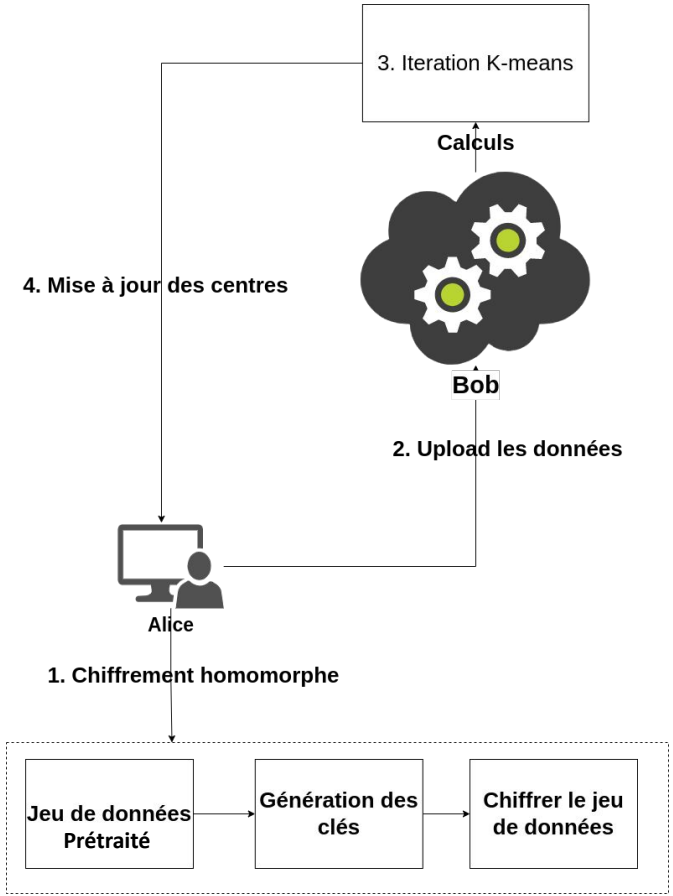
Bootstrapping programmable et fonctionnel



Conception

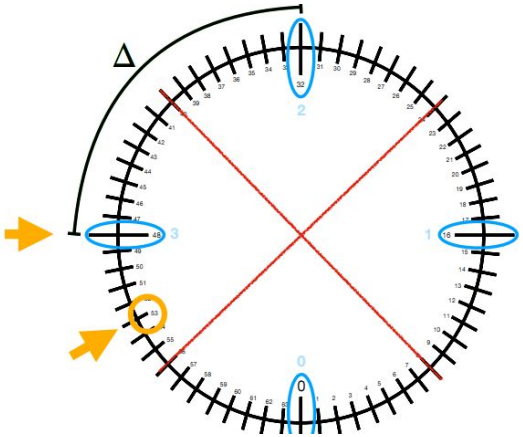
Architecture de K-MEANS-HE

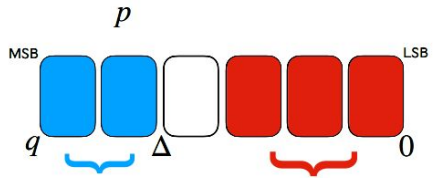
- **La génération des clés se fait au niveau de propriétaire de données.**
- **Le propriétaire des données encode et chiffre les données.**
- **L'étape d'affectation s'effectue au niveau de serveur cloud sans déchiffrement intermédiaire.**
- **La division est laissée au niveau du propriétaire de données.**



Prétraitements

- **Effectuer une standardisation Z-score sur le jeu de données.**
- **Effectuer une normalisation de type Min-Max sur le jeu de données.**
- **Encoder les données sur un tore.**





$\mathcal{M} = \{0,1,2,3\}$

$\text{Encode}(m) = \Delta m$

$|e| < \frac{\Delta}{2} = 8 = 2^3$

Example: $m = 3$, $\Delta m = 48$, $e = 5$

$\Delta m + e = 53$

KMEANS proposé en clair

- **Version proposée de k-means**

Input : X, k
Output : C, A
Initialisation
while *non convergence* **do**
  Affectation : for $x_i \in X$ **do**
    Construction de la matrice Δ_i
    Calcul de vecteur d'affectation $A^{(i)}$
  Mise à jour des centres

- **Les principales modifications se situe à l'étape d'affectation pour permettre d'effectuer la comparaison de manière sécurisée et sans déchiffrement intermédiaire.**

Affectation d'un point a (1)

- **Construction de la matrice Δ :**
  - **Calculer les différences des distances entre le point a et les centres deux à deux.**

$$d_{ai}^2 - d_{ai'}^2 = \sum_{j=0}^{d}(C_{ij}^2 - C_{i'j}^2) + -2 * \sum_{j=0}^{d}(C_{i'j}^2 - C_{ij}^2)a_j$$

  - **Appliquer la fonction signe.**

$$\begin{pmatrix} \delta_{0,0} & \delta_{0,1} & \cdots & \delta_{0,k-1} \\ \delta_{1,0} & \delta_{1,1} & \cdots & \delta_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{k-1,0} & \delta_{k-1,1} & \cdots & \delta_{k-1,k-1} \end{pmatrix}$$

$$\delta_{i,j} = \begin{cases} 1 & \text{si } d_{ai}^2 < d_{aj}^2 \\ 0 & \text{sinon.} \end{cases}$$

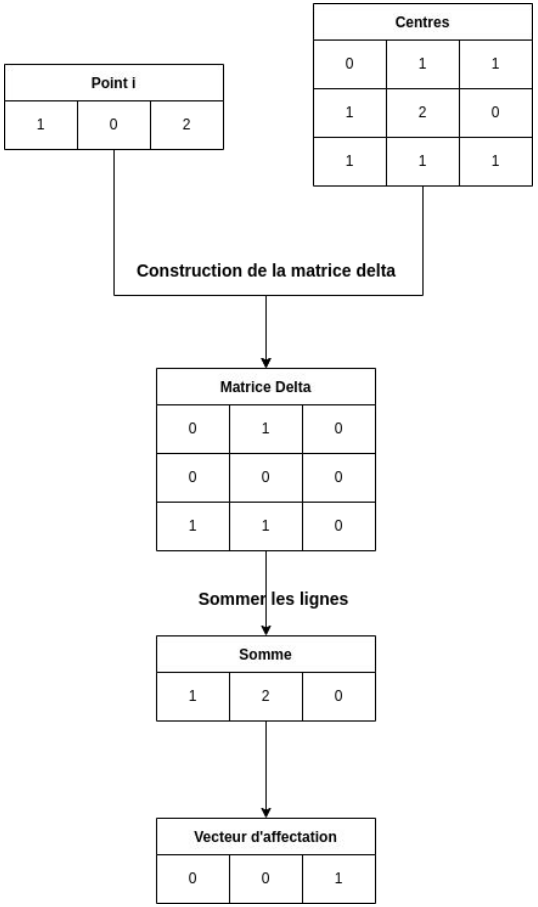
Affectation d'un point a (2)

- **Calcul de vecteur d'affectation :**
  - **Théoriquement, il suffit de faire la somme des lignes de la matrice Δ avoir un index de tri des distances.**

$$(\Delta_0, \Delta_1, ..., \Delta_{k-1}) \text{ avec } \Delta_j = \sum_{i=0}^{k-1} \delta_{i,j}$$

  - **Il suffit ensuite d'appliquer la fonction suivante**

$$\overline{sign}(\Delta_i) = \begin{cases} 1 & \text{si } \Delta_i \leq 0 \\ 0 & \text{sinon.} \end{cases}$$



De la version en clair vers TFHE (1)

- **Contraintes :**
  - **La multiplication ne se fait que par un scalaire non chiffré.**
  - **Le domaine de définition des valeurs et des résultats est le tore.**
  - **Nécessité de gérer le bruit.**

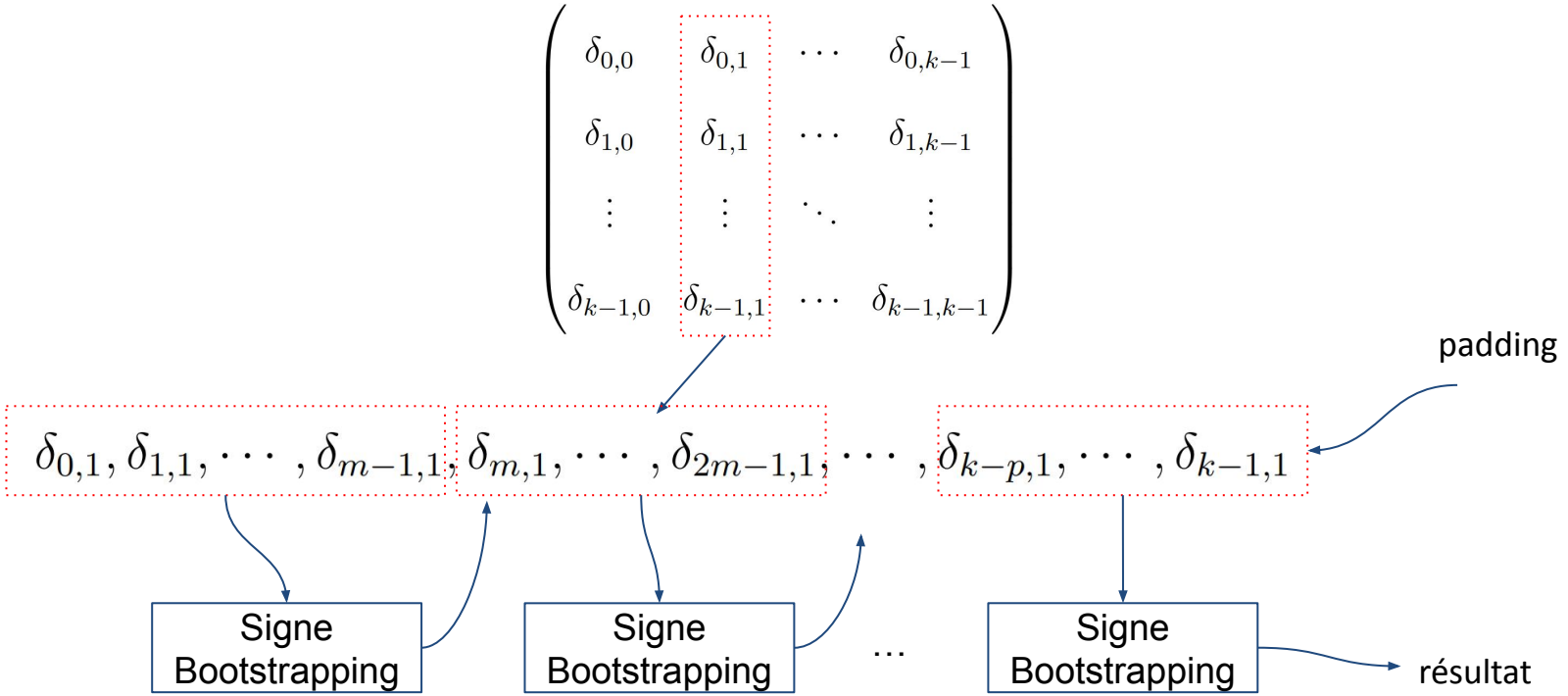
- **Verrous :**
  - **Quels sont les données à laisser en clair ?**
  - **Comment garder la précision des scalaires ?**
  - **Comment encadrer les résultats intermédiaires ?**
  - **Comment gérer le bruit en calculant la somme ?**

De la version en clair vers TFHE (2)

- **Laisser les centres en clair et les arrondir.**
- **Multiplier les centres par un paramètre de précision τ.**
- **Diviser les données par le paramètre de précision τ.**
- **Diviser les données et les centres par un facteur de cadrage v.**
- **Implémenter la fonction signe en utilisant le bootstrapping programmable.**

$$\sum_{j=0}^{d} Round(\frac{\tau \times c_{ij}}{v}) \times \frac{x_{ij}}{\tau v}$$

De la version en clair vers TFHE (3)



K-PODS

- **Essaie de minimiser la fonction :** $min_{c_1,\ldots,c_K,C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{ij, i \in C_k \text{ et } ij \in O} \left(z_{ij} - c_{kj} \right)^2$
- **Pas de solution triviale pour minimiser cette fonction.**
- **Utiliser un algorithme majoration minimisation pour la minimiser.**
- **L'algorithme k-pods suit l'allure suivante :**

1. Initialisation.
2. Répéter jusqu'à convergence {
   Majoration : compléter la matrice initiale.
   Minimisation : exécuter k-means.
   }

Prise en compte des données manquantes

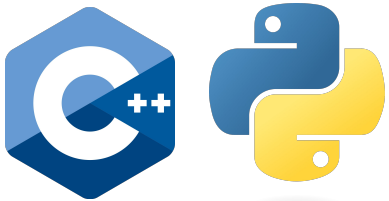
- **Il suffit d'introduire l'algorithme k-means-HE dans l'étape de majoration**

1. Initialisation.
2. Répéter jusqu'à convergence {
   Majoration : compléter la matrice initiale.
   Minimisation : exécuter k-means-HE.
   }

- **L'étape d'initialisation consiste à initialiser les centres et la matrice d'affectation A.**
- **L'étape de majoration doit être effectué en clair pour éviter les fuites d'information.**

Technologies utilisées

Logiciels utilisés.



Langages de programmation

Bibliothèques utilisées

TESTS ET RESULTATS

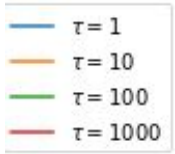
Test sur TFHE

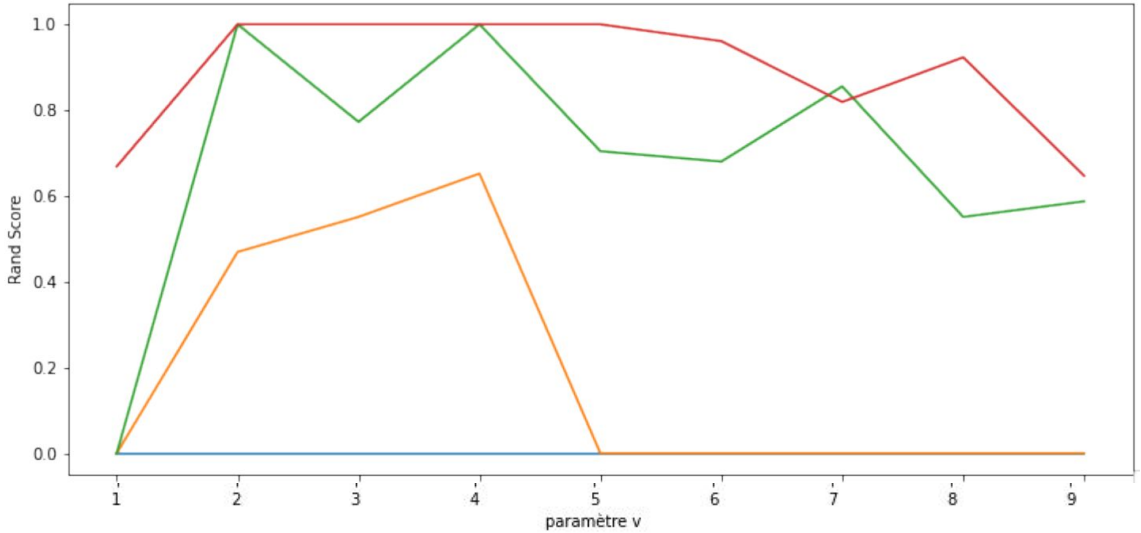
- **Le but est d'expliquer le choix de la dimension du problème LWE.**
- **Selon les recommandations des concepteurs, actuellement, il faut avoir des paramètres qui offre une sécurité de plus de 128 bits.**
- **Lwe-estimator pour calculer le niveau de sécurité offert par les paramètres.**

Travaux	N	σ	λ
(Chillotti et al.,2016)	1024	2^{-25}	129
(Zuber, 2020)	1024	10^{-9}	108
Ce travail	2048	10^{-15}	134

Choix des paramètres de Kmeans-HE

- **Le but est de confirmer les hypothèses théoriques sur les paramètres.**
- **Le jeu de données utilisé est Iris.**





Tests de performances (Efficacité)

- **Évaluation interne et externe de k-means-HE sur différents jeux de données.**

Jeux de données	Nb iter (clair)	ARI
Iris	5 (5)	1.0
Wine	9 (10)	0.98
Glass	10 (10)	0.97
Ovarian	4 (4)	1.0
Breast Cancer	9 (9)	1.0
Mice Protein	17 (15)	0.92
Pen digits	30 (30)	0.93

Jeux de données	kmeans-HE		sklearn	
	Inertie	Sil	Inertie	Sil
Iris	6.98	0.5	6.98	0.5
Wine	48.96	0.30	48.97	0.299
Glass	19.22	0.365	19.20	0.365
Ovarian	367.04	0.439	367.04	0.439
Breast Cancer	215.83	0.385	215.83	0.385
Mice Protein	971.42	0.135	971.13	0.136
Pen digits	3594.47	0.28	3584.97	0.28

Tests de performances (Rendement)

- **Temps d'execution**

Jeux de données	n	d	Nb classes
Iris	150	4	3
Wine	178	13	3
Glass	214	9	6
Ovarian	216	100	2
Breast Cancer	699	9	2
Mice Protein	1077	77	8
Pen digits	10992	16	10

Nb iter	Sequentiel	iteration /individu	8 threads
5	80s	0.10s	19s
9	175s	0.10s	59s
10	2853s	1.33s	250s
4	69s	0.08s	19s
9	183s	0.03	55s
17	15187s	0.94s	6289s
30	531222s	1.61s	82958s

Tests de performances (Sécurité)

- **Les paramètres offre une sécurité qui dépasse 128bits.**
- **Fuite d'information au niveau des centres des données.**
- **Les centres peuvent être protégés par d'autres méthodes de cryptographie telle que la confidentialité différentielle.**
- **Dans les hypothèses de ce travail, la solution peut être considérée sûre dans un modèle semi-honnête.**

Tests sur kPODS

- **Utiliser l'indice de rand ajusté.**
- **Comparer les résultats obtenus par kPODs implémenté sur python et notre version proposée.**
- **L'indice de rand est calculé par rapport à la version claire de k-means.**

Jeux de données	Taux d'absence	ARI (ce travail)	ARI (chi et al., 2016)
Iris	10%	0.80	0.78
Wine	30%	0.62	0.64
Glass	20%	0.57	0.54
Ovarian	10%	0.45	0.46
Breast Cancer	10%	0.71	0.78

Conclusion

Contribution

- **Proposition d'une version de k-means en utilisant le chiffrement homomorphe.**
- **Effectuer l'ensemble de l'étape d'affectation sur le serveur distant.**
- **Implémentation de la version k-means-HE.**
- **Augmentation de la solution pour prendre en compte les données manquantes grâce à la méthode des kPODs.**

Perspectives

- **Améliorer la sécurité des centres grâce à d'autres techniques.**
- **Prévoir d'autres tests pour la partie données manquantes.**
- **Explorer d'autres techniques de tri dans le but d'améliorer la vitesse d'exécution.**
- **Adapter d'autres algorithmes d'apprentissage automatique dans le contexte HE.**

DEMONSTRATION

Démonstration

- **Montrer que les paramètres utilisés sont sûrs grâce à lwe-estimator.**
- **Utiliser les jeux de données iris et wine.**
- **Exécuter kmeans-HE sur notre machine locale.**

Bibliographie

- Chi, J. T., Chi, E. C., and Baraniuk, R. G. (2016). k-pod : A method for k-means clustering of missing data. The American Statistician, 70(1) :91–99.
- Chillotti, I., Gama, N., Georgieva, M., and Izabachene, M. (2016). Faster fully homomorphic encryption : Bootstrapping in less than 0.1 seconds. In international conference on the theory and application of cryptology and information security, pages 3–33. Springer.
- Forgy, E. (1965). Cluster analysis of multivariate data : Efficiency versus interpretability of classification. Biometrics, 21(3) :768–769.
- Marbac, M., Sedki, M., and Patin, T. (2020). Variable selection for mixed data clustering : application in human population genomics. Journal of Classification, 37(1) :124–142.
- Zuber, M. (2020). Contributions to data confidentiality in machine learning by means of homomorphic encryption. PhD thesis. Thèse de doctorat dirigée par Sirdey, Renaud Mathématiques et Informatique université Paris-Saclay 2020.

Merci pour votre attention…

QUESTIONS

Le problème LWE

- **S**i s est un vecteur secret, alors il est aisé de retrouver s étant donné des produits scalaires <a,s>, si l'on connaît suffisamment de vecteurs a. Cela se résout facilement par un pivot de gauss.
- Ajouter une erreur permet de rendre le problème difficile.

Algorithmes de bootstrapping

Algorithme 5 : Opération Sign

Input : un TLWE Sample $(a, b) = [m]$ du message μ,
une clé de bootstrapping BK, une base de bootstrapping b_δ

Output : $LWE(\frac{1}{(b_\delta)})$ si $m + \frac{1}{b_\delta} \in [0, \frac{1}{2}]; 0$ sinon

Let $\bar{b} = \lfloor 2Nb \rceil$

for $i = 1$ *to* n **do**

  $\bar{a_i} = \lfloor 2Na_i \rceil$

Let $testv = (1 + X + \cdots + X^{N-1} \times X^{-\frac{2N}{4}} . \frac{1}{2 \times b_\delta})$

ACC $\leftarrow [X^{\bar{b}}, (0, testv)]$

for $i = 1$ *to* n **do**

  ACC $\leftarrow [\, h + (X^{\bar{a_i}} - 1).bk_i)] \; .ACC$

return $SampleExtract(ACC) + [\frac{1}{2 \times b_\delta}]$

Majoration minimisation