Machine-Learning-as-a-service (MLaaS) confidentielle basé sur le chiffrement homomorphe – Enjeux et Verrous

Charly Bechara, Santiago Cortijo, Olivier Heron

Machine-Learning-as-a-service (MLaaS) confidentielle basé sur le chiffrement homomorphe – Enjeux et Verrous

Privacy preserving Machine-Learning-as-a-service (MLaaS) based on homomorphic encryption – Issues and Challenges

Charly BECHARA
SCALNYX SAS
Paris, France
charly.bechara@scalnyx.com

Santiago CORTIJO
SCALNYX SAS
Paris, France
santiago.cortijo@scalnyx.com

Olivier HERON
CEA, LIST,
91191 Gif-sur-Yvette Cedex, France
olivier.heron@cea.fr

***Résumé*—Le chiffrement homomorphe appliqué au MLaaS garantit la confidentialité des données sur toute la chaîne de traitement. Cette communication vise à relever les verrous technologiques, et à proposer une solution architecturale à haute performance et à moindre coût.**

***Mots clefs*—MLaaS, chiffrement homomorphe, FHE, haute performance, programmation parallèle, compilateur**

***Abstract*—Homomorphic encryption applied to MLaaS guarantee the data privacy throughout the whole processing chain. This article aims to address the various technological obstacles, and to propose a high-performance architectural solution at lower costs.**

***Index terms*—MLaaS, homomorphic encryption, FHE, high performance, parallel programming, compiler**

I. INTRODUCTION

Dans le cadre d'un projet d'évaluation commun pour relever le défi de la conception de l'intelligence artificielle de confiance à l'aide du chiffrement homomorphe et son passage à l'échelle sur des machines Cloud, le CEA et SCALNYX se sont associés via une lettre d'intention pour étudier conjointement la faisabilité technique et pour lever les verrous afin de développer un prototype industriel. La coopération a débuté en Novembre 2019.

Selon une étude menée par Research and Markets, le marché des Machine Learning as a Service (MLaaS) devrait rencontrer une croissance de 49% durant la période de 2017 à 2023 [1], et son principal avantage est de permettre aux PMEs d'accéder aux mêmes technologies de Machine Learning que les grandes sociétés. Malheureusement, dans le cadre traditionnel du MLaaS il y a un compromis à faire entre la confidentialité des données et l'information que l'on peut en extraire, ce qui est inacceptable pour certaines applications qui contiennent des données sensibles.

Le développement des environnements hébergés (Cloud) amène des défis de sécurité et de confidentialité dont les réponses s'appuient sur le choix d'un principe : soit l'établissement de barrières autour des

données, solution coûteuse et qui comporte toujours des risques, soit le cryptage des données elle-mêmes, ce qui garantit par-design le niveau de sûreté indépendamment de l'environnement, mais interdit tout traitement sur ces données in-situ.

Grâce au chiffrement homomorphe, effectuer des calculs sur ces données cryptées sans connaître leurs valeurs est maintenant possible ce qui ouvre un champ d'opportunités étendu pour garantir la confidentialité des données hébergées sur des serveurs publics ; les données sont manipulées uniquement sous forme chiffrée et le résultat des opérations n'est accessible qu'avec la clef de déchiffrement.

Développée par le CEA, au sein de l'institut List, la technologie Cingulata [2] est une chaîne logicielle de compilation unique permettant de créer des applications capables d'effectuer des traitements sur des données chiffrées. Cette technologie rend accessible ce domaine de la Cryptographie aux développeurs d'application et intègre des optimisations pour la performance. Par ailleurs, Cingulata intègre notamment le cryptosystème homomorphe TFHE (FHE on Torus) qui est l'algorithme de chiffrement le plus performant à ce jour pour des calculs au niveau du bit de l'information.

Les applications de type MLaaS (Machine Learning as a Service) sont adoptées dans plusieurs domaines industriels. L'utilisation du chiffrement homomorphe pour les applications, par exemple de type réseaux de neurones permettra aux solutions MLaaS de faire l'entraînement et l'inférence sur des données "sensibles" tout en garantissant la confidentialité entre les acteurs. Cependant, le passage à l'échelle sur des machines Cloud avec un rapport performance/prix acceptable économiquement reste un défi et un enjeu important dans l'industrie.

Dans ce contexte, il devient nécessaire et indispensable de pouvoir amener une rupture technologique qui permette la construction d'une solution dédiée, avec une méthodologie de conception matérielle et logicielle, avec une garantie forte de scalabilité permettant des traitements massivement parallèles.

II. MLaaS

Nous sommes maintenant à un point où l'informatique haute performance (HPC) et le cloud travaillent ensemble pour alimenter des solutions qui amènent les entreprises à innover. Le HPC dans le cloud est une option de plus en plus viable pour les organisations de toutes tailles et ne fera que se renforcer à mesure que les technologies continuent de converger et de mûrir. L'intelligence artificielle (IA) et l'apprentissage automatique (ML) se déplaceront de plus en plus dans le cloud.

L'IA et le ML peuvent nécessiter beaucoup de calculs, en particulier pour l'entraînement, qui peut impliquer d'énormes quantités de données. Le cloud aide les organisations à répondre aux exigences de calcul sans avoir besoin de leurs propres centres de données. Les organisations cherchant à générer de la valeur commerciale avec l'IA et le ML bénéficieront des avantages en termes de prix et de performances avec le cloud. Elles s'appuieront également sur des solutions d'orchestration des processus pour faciliter la convergence des charges de travail ML à forte intensité de calcul à travers l'infrastructure HPC. Nous pressentons l'émergence d'une demande pour une plate-forme unique qui peut fournir un accès dynamique à la technologie de simulation et aux ressources informatiques évolutives qui comprendra le traitement des données, voire les données elles mêmes. Les entreprises s'orientent vers l'approvisionnement dynamique des ressources informatiques au lieu de s'appuyer sur des applications rigides qui vivent à plein temps dans un environnement informatique prédéterminé.

La visibilité et l'utilisation du HPC se développent au sein des organisations et elles seront adoptées dans un plus grand nombre d'équipes au sein des organisations pour rationaliser les processus et la production. Les équipes exploiteront le HPC pour explorer plus d'opportunités en moins de temps et battre les concurrents sur le marché avec de nouvelles solutions.

A. Contraintes de réglementation

Les applications MLaaS industrielles où les traitements sur des données sensibles sont indispensables, doivent répondre également à des contraintes qui sont parfois conflictuelles telles que : sécurité, sûreté de fonctionnement, obligations réglementaires, temps réel, économie de machines et de consommation énergétique ; le RGPD (Règlement Général sur la Protection des Données) couplé avec des normes spécifiques comme MiFID II pour l'industrie financière, ou bien la loi bioéthique pour la santé ; l'industrie du transport, de la défense et de l'industrie du futur sont soumises à des normes de sûreté de fonctionnement et de sécurité. Par exemple, dans le domaine automobile, la sûreté de fonctionnement est actuellement régie par le standard ISO 26262, qui permet d'établir un cadre et des exigences sur les systèmes embarqués dans le véhicule afin de garantir leur sécurité.

B. Contraintes de confidentialité

La confidentialité des donnés est explicitement affichée dans des nombreux textes législatifs dont la loi Informatique et Libertés de 1978 et la loi sur le secret bancaire (loi de 1984) qui ont imposé les plus hauts degrés de sécurité. Désormais, une amende de la Commission nationale informatique et libertés (Cnil) peut atteindre 4% du chiffre d'affaires mondial de l'établissement fautif. Et le gendarme des banques, l'Autorité de contrôle prudentiel et de résolution (ACPR) peut y ajouter une pénalité de 100.000 euros à laquelle s'ajoutent encore les sanctions pénales liées aux violations du secret professionnel, délit sanctionné d'un an d'emprisonnement et de 15.000 euros d'amende. Les banques ont aussi dû expliciter et publier leurs politiques de protection de données, en accès libre sur leurs sites, et leurs conventions de compte. Elles doivent désormais demander aux clients leur consentement pour les profilages réalisés en vue d'une prospection commerciale en agence, sur Internet ou dans leur application, au lieu de les demander à la Cnil.

C. RGPD

Entré en vigueur le 25 mai 2018, le Règlement général sur la protection des données (RGPD) est applicable dans les 28 pays membres de l'Union européenne, et vise à harmoniser la réglementation en matière de protection des données personnelles. Sa mise en application est complexe et représente un défi pour le secteur de la banque qui gère un grand nombre de données particulièrement sensibles. Le dispositif octroie de nouveaux droits aux consommateurs, dont les droits à l'accès, la portabilité, la rectification et l'oubli. La portabilité des données constitue quant à elle un nouveau droit permettant aux clients de récupérer et de disposer de leurs données personnelles comme ils le souhaitent. Ils peuvent par exemple les transmettre à une institution bancaire ou une *fintech* pour souscrire à différents services. En cas d'incident, il oblige aussi les établissements à prévenir les autorités sous soixante-douze heures en leur indiquant la manière dont elles vont résoudre les problèmes et avertir les personnes concernées. Le RGPD crée des contraintes majeures dans les systèmes informatiques, sur des informations à haute valeur et à haut risque qui nécessitent des changements décisifs des méthodes de traitement des données dans les établissements financiers.

III. Cryptographie Homomorphe et Machine Learning

Le chiffrement homomorphe est un concept mathématique qui répond au problème de la sécurisation des données. Pour expliquer son bénéfice, supposons un scénario à trois acteurs : Alice et Bob ont des données privées et Charlie peut faire quelque chose d'utile pour Alice en utilisant à la fois les données d'Alice et de Bob. Le chiffrement homomorphe permet à Charlie d'utiliser les données de Bob et Alice mais sans qu'elles lui soient révélées. En complément des opérations classiques de chiffrement et de déchiffrement des données, le chiffrement homomorphe permet d'effectuer tous les types de calcul possibles dans le domaine chiffré, sans disposer du résultat intermédiaire ni final du calcul. Sur la base de ce paradigme de chiffrement, il est possible de construire un modèle simple de confiance qui implique deux parties : un utilisateur, propriétaire des données privées, et un serveur, propriétaire d'un algorithme qui s'exécute sur une machine (non-de-confiance). Les données sont tout d'abord chiffrées par l'utilisateur puis transmises en chiffré au serveur. Le serveur exécute le traitement avec les données reçues, voire peut également injecter des données qui lui sont propres dans le traitement. Le résultat est chiffré, puis est retourné à l'utilisateur qui peut récupérer le résultat après déchiffrement. Le serveur ne dispose des données en clair de l'utilisateur à aucun moment. D'autres modèles de confiance à plusieurs utilisateurs sont possibles pour adresser un plus large éventail de cas d'usage.

A. Etat de l'art

Vers la fin des années 70 apparaît un article historique de Rivest, Adleman et Dertouzos [3], qui définissent et étudient le potentiel applicatif d'une nouvelle notion qu'ils appellent l'homomorphisme confidentiel. En effet, en s'appuyant sur le fait fondamental que le cryptosystème RSA permet la multiplication en homomorphe - le produit de deux textes chiffrés fournit un chiffrement du produit des deux textes clairs correspondants - ils finissent par conjecturer l'existence de cryptosystèmes permettant d'effectuer des calculs directement sur des données chiffrées. Cette idée restera une curiosité pendant plusieurs années. Cette situation a changé vers la fin des années 90, lorsque, principalement en raison de l'introduction du cryptosystème Paillier [4], la recherche de cryptosystèmes homomorphes en même temps pour les opérations d'addition et de multiplication (ce que l'on appelle le chiffrement totalement homomorphe - FHE) devient l'une des quêtes du Graal d'une partie

de la communauté cryptographique. En 2009, contre toute attente, C. Gentry, alors à Stanford, propose une première construction crédible d'un schéma de chiffrement totalement homomorphe à la fois en termes de sécurité et d'efficacité théorique [5]. Pour résoudre le problème lié à l'amplification du bruit qui se cumule lors de chaque opération (surtout pour la multiplication), Gentry a introduit une technique de réduction de bruit, connue sous le nom *bootstrapping*, qui consiste en gros à effectuer de manière homomorphe une opération de rechiffrement sans jamais avoir les données sous format clair pendant cette opération. Malheureusement, ce premier cryptosystème totalement homomorphe et jusqu'à récemment tout cryptosystème basé sur le *bootstrapping*, restent beaucoup trop coûteux pour avoir une pertinence pratique. Les choses se sont améliorées par la suite vers 2011 lorsque [5] (à nouveau) et deux coauteurs (Z. Brakerski et V. Vaikuntanathan) ont proposé un nouveau schéma : les schémas totalement homomorphes par niveau [6]. Ces schémas de chiffrement, dits presque homomorphes (SHE), sont dus à Z. Brakerski [7] ainsi qu'à Fan et Vercauteren [8], puis optimisés dans [9] et [10] pour aboutir au schéma BFV. Une troisième génération de schémas, inventés à partir des schémas FHE de 2ème génération, tels que [11] ont également été proposés, mais bien qu'ils soient conceptuellement plus simples que les SHE par niveau, ils semblent moins efficaces. Puis, à partir de 2015, une nouvelle génération plus rapide de schémas FHE basé sur le *bootstrapping* a commencé à apparaître [12], [13], [14], notamment TFHE (*Torus-based FHE*) qui est dérivée d'une réinterprétation de GSW et de ses variantes d'anneaux et à la conception duquel le CEA LIST a contribué. De plus, des schémas FHE pour manipuler l'arithmétique approximative ont été proposés, notamment HEAAN [15], [16]. Tout récemment, HEEAN, BFV et TFHE ont été unifiés dans un cadre théorique cohérent connu sous le nom de Chimera [17] qui permet de passer de manière homomorphe d'un cryptosystème à l'autre et, par conséquent, de choisir le schéma le plus approprié pour effectuer une partie des calculs. Alors que la performance des schémas généralistes FHE atteignent un plateau (hors accélération matérielle [18], [19]), il apparaît un champs d'opportunités pour développer des optimisations spécifiques de la performance au niveau applicatif. Concernant le surcoût dans la communication et du stockage des données (car les données chiffrées sont intrinsèquement beaucoup plus grandes que leur homologue en texte clair), il existe maintenant plusieurs algorithmes de chiffrement symétriques qui peuvent être appliqués efficacement sur des schémas FHE [20], [21], [22], [23], et permettre alors d'exploiter des capacités de transchiffrement pour résoudre ce problème.

B. CEA LIST Cingulata

Sur le plan de la programmation, des chaînes d'outils de compilation permettant d'écrire, d'optimiser automatiquement puis d'exécuter des programmes complexes écrits dans des langages de haut niveau standard (par exemple C++ et Python) sont désormais opérationnels [24], [25], [26].

Dans ce contexte, Cingulata [2] est le premier compilateur open-source pour les applications homomorphes écrites en C++, un langage de codage simple et de haut niveau, développé et maintenu par le CEA LIST. Dans la partie frontale de Cingulata, les développeurs disposent d'un type d'entiers spécifique, appelé CiInt, pour déclarer les variables chiffrées utilisées dans leur application. Ensuite, après la compilation, Cingulata génère un circuit booléen qui manipule des bits chiffrés. L'utilisation d'un circuit booléen convient aux principaux systèmes de chiffrement homomorphe car ces systèmes chiffrent les bits séparément (ces systèmes de chiffrement sont par exemple BGV, BFV et TFHE avec «gate-bootstrapped»).

La version 1 de Cingulata implémente le cryptosystème BFV (schéma de chiffrement homomorphe par niveau). Les paramètres de sécurité BFV, et par conséquent les performances d'exécution, dépendent de la profondeur multiplicative de la fonction qui sera évaluée dans le domaine homomorphe. La profondeur multiplicative d'un circuit se réfère au nombre maximum de portes ET sur n'importe quel chemin d'une entrée à une sortie du circuit. La profondeur multiplicative est minimisée lors de la compilation de l'application grâce à l'utilisation de scripts d'optimisation personnalisés et de l'outil ABC [27]. L'environnement d'exécution Cingulata permet d'exécuter le circuit booléen de l'application. Cet environnement fournit des commandes pour chiffrer, déchiffrer et évaluer des circuits avec des entrées chiffrées. La figure 2 décrit le processus de compilation complet de Cingulata v1.

La version 2 du compilateur Cingulata a été publiée en 2019. Elle prend en charge un cryptosystème «gate-bootstrapped» appelé TFHE. TFHE est considéré comme l'algorithme de chiffrement homomorphe «gate-bootstrapped» le plus rapide. Cette version permet de sélectionner le cryptosystème BFV ou TFHE. Par conséquent, les développeurs peuvent tester facilement leurs codes réalisés avec la version 1 et la version 2 de

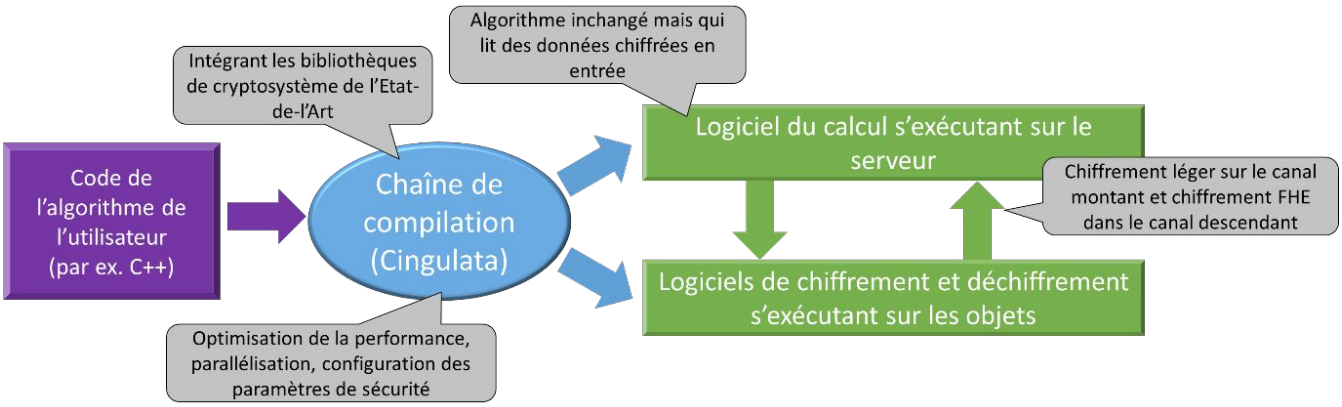

FIGURE 1. Principe de Cingulata.

Cingulata. La compatibilité descendante avec la version 1 est assurée dans cette nouvelle version.

Cingulata supporte des tailles différentes de données grâce à son type d'entiers CiInt qui est paramétrable. Par ailleurs, le compilateur expose des structures de vecteurs de chiffrés pour faciliter l'implémentation d'algorithme de chiffrement symétrique (par ex. Trivium). Cingulata implémente des opérateurs optimisés spécifiquement pour des opérations effectuées avec BFV et TFHE. Chacun des deux cryptosystèmes a des objectifs différents. Basé sur le cryptosystème BFV, l'objectif est de réduire la profondeur multiplicative du circuit booléen. Cingulata implémente des opérateurs addition, multiplication et fonction de multiplexage exposant une profondeur réduite. Basé sur le schéma TFHE, l'objectif est de réduire le nombre de portes du circuit booléen. Cela amène un choix différent dans l'implémentation des opérateurs qui minimise le nombre de portes, par exemple additionneur à propagation de retenue. Cette approche de conception facilitera l'intégration d'autres cryptosystèmes, tels que SEAL et HELIB. Enfin, Cingulata embarque une collection de tests unitaires, basée sur la suite Googletest (https ://github.com/google/googletest), des exemples d'application simple et de la documentation qui couvre la description des méthodes publiques.

Des standards émergent lentement pour le FHE via l'initiative https ://homomorphicencryption.org à laquelle contribue l'équipe du CEA LIST. Plus important encore, certains travaux, pour lesquels le groupe CEA LIST a été à l'avant-garde, ont pu démontrer que le chiffrement homomorphe pouvait être intégré avec succès dans des systèmes réels et conduire à des performances conformes aux contraintes de latence des applications dans des domaines tels que le diagnostic médical, le génomique,

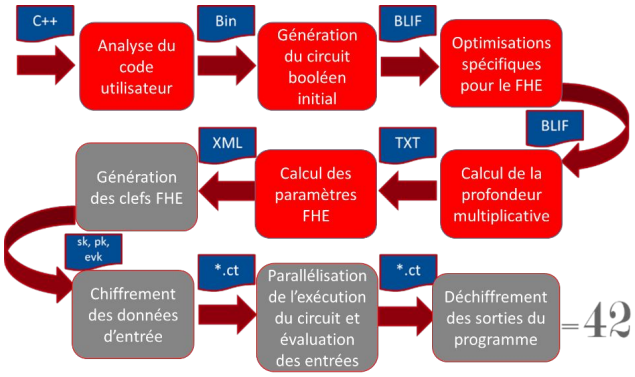

FIGURE 2. Compilation d'une application en homomorphe avec Cingulata.

l'authentification biométrique, la classification de base et quelques autres [28], [29], [30], [19], [31], [32], [33], [34].

C. Machine learning et FHE

Relativement au domaine du « machine-learning », et plus précisément celui relatif aux réseaux de neurones artificiels (ANN), la recherche sur l'application de techniques de calcul sur des données chiffrées, FHE ou d'autres techniques concurrentes, pour résoudre le problème de confidentialité des ANN n'est qu'à ses débuts et n'a jusqu'à présent qu'à peine effleuré la surface du problème. Il existe, au moment de la rédaction de ce document, une dizaine d'articles sur ce sujet, dont beaucoup en prépublications, et tous publiés entre 2016 et aujourd'hui. Jusqu'à présent, les premières tentatives d'application de techniques de chiffrement homomorphe aux ANN se sont toutes concentrées uniquement sur la phase d'inférence et plus spécifiquement sur le problème de l'évaluation d'un réseau en clair (du point de vue

de l'ordinateur effectuant l'évaluation) sur une entrée chiffrée (produisant ainsi une sortie cryptée) [35], [36], [37], [38], [39], [40], [41], [42], [43], [44]. Par ailleurs, les travaux publics sur l'application du FHE pendant la phase d'apprentissage et sur des techniques de réseau neuronal autres que les réseaux de neurones convolutifs sont inexistants, seuls certains travaux se concentrant sur le regroupement de base [45], [46] et certains sur l'apprentissage de modèles de régression logistique [47], [48].

IV. Analyse des verrous de la Cryptographie Homomorphe avec MLaaS

Afin d'identifier les verrous actuels de l'utilisation de la cryptophtahie homomorphe (FHE), nous nous sommes principalement basés sur les expériences suivantes :

- Exploration d'accélérations de l'outil Cingulata à travers l'intégration avec un runtime d'exécution sur des CPUs multicoeurs généralistes pour permettre le passage à l'échelle : ordonnancement et parallélisme efficace de l'exécution des opérateurs homomorphes.
- Portage "manuel" d'un modèle de réseaux de neurones de type MLP (multi layer perceptron), avec des poids synaptiques publiquement connus, ayant entrées/sorties encryptées, en analysants ses optimisations HPC (bonne utilisation de cache, non-redondance des calculs).

A. Analyse applicative

- En fonction de la complexité de l'algorithme, certains cryptosystèmes sont mieux adaptés que d'autres. Ceci dit, un framework permettant l'exploration rapide du meilleur cryptosystème pour une application cible est nécessaire.
- Les bibliothèques mathématiques encryptées de base pour construire des réseaux de neurones n'existent pas encore. Les opérateurs de base d'addition et de multiplication existent, mais il reste à explorer des solutions pour faire des fonctions plus complexes comme la division, la convolution, les fonctions exponentielles, trigonométriques, ainsi que des fonctions nécessaires de l'algèbre linéaire.
- L'automatisation de l'intégration et du déploiement des systèmes de chiffrement homomorphe est primordiale pour une plateforme MLaaS (Machine Learning as a Service). Ce processus est aujourd'hui manuel, peu efficace, et très sensible à des erreurs humaines qui pourront impacter la sécurité des données et les performances globales.
- Plateforme logicielle collaborative : la technologie est accessible uniquement par les experts de la cryptographie. Cette plateforme vise à permettre à des industriels utilisateurs de cette technologie de monter en connaissance et d'évaluer son bénéfice pour leurs cas d'usages avant d'investir pour un usage au quotidien.

B. Analyse de performance

- Le compilateur Cingulata génère des circuits booléens dont la taille et la profondeur multiplicative dépendent de la complexité et de la nature du code donné en entrée. Faire du machine learning et du deep learning pourra rentrer dans une catégorie qui nécessite un nombre de CPUs et une taille mémoire très élevée pour effectuer les traitements homomorphes.
- Le temps de calcul de l'algorithme avec des données encryptées est plus élevé qu'une exécution sur des données en clair. A notre connaissance, il n'existe pas encore de solution de chiffrement homomorphe dans l'État de l'Art permettant d'adresser des applications temps-réels qui nécessitent des performances déterministes et basse latence.
- La taille des données après chiffrement est augmenté d'un facteur entre 10^3 à 10^6 en fonction du niveau de sécurité visé. Néanmoins, les performances du calcul vont dépendre de la qualité de gestion des données dans la mémoire et de la proximité des calculs par rapport aux données. Ce sont des problèmes de type "in-memory computing".
- Les opérateurs mathématiques de multiplication et d'addition sont essentiellement portés sur des cibles de CPU x86. Il existe une réalisation de portage du TFHE sur GPU dans l'État de l'Art [49]. Il y a certainement des opportunités à explorer d'autres cibles d'architectures, par exemple le MPPA de Kalray [50] et ARM64.
- L'efficacité énergétique des calculs n'est pas encore prise en considération dans la conception d'une implémentation utilisant le chiffrement homomorphe, mais pourrait être un critère important au moment de choisir une architecture.

V. Solution Architecturale

Nous souhaitons dans un premier temps répondre aux verrous de performance en proposant une solution architecturale adaptée et scalable par conception. En

effet, pour trouver une solution optimale pour faire de la cryptographie homomorphe sur MLaaS, il faut bien séparer les différents domaines de conception et regarder au delà du domaine algorithmique : certains calculs peuvent être réordonnés sans changer les résultats numériques, mais en changeant radicalement ses performances, et ces optimisations sont en général très dépendantes du matériel sur lesquels elles sont exécutées (une optimisation conçue pour un certain processeur peut facilement être contre-productive sur un autre).

Dans ce contexte, afin de déployer un système MLaaS maintenable et capable d'évoluer (avec des nouveaux algorithmes et sur des architectures différentes), il devient nécessaire de découpler les représentations algorithmiques des optimisations de performance. Pour lever ce verrou, nous nous appuierons sur la méthodologie AAA (Adéquation Algorithme Architecture) et l'adapterons à notre problématique. La méthodologie AAA est une discipline assez réputée dans le monde de l'embarqué pour concevoir des architectures matérielles dédiées à une application/algorithme particulier. C'est un problème d'optimisation qui consiste à choisir une implémentation dont les performances respectent les contraintes temps-réels, énergétiques et surfaciques. L'algorithme est modélisé sous forme de graphe de flots de données, permettant la capacité d'expression de tout le parallélisme, et d'effectuer ainsi des optimisations précises prenant en compte la conception conjointe logiciel/matériel (« co-design ») et de simplifier la génération de code.

Les optimisations algorithmiques sont gérées par Cingulata [2], et les optimisations de performance seront gérées par une nouvelle chaîne de compilation dédiée pour la performance. Le nouveau compilateur SCALGO (développé par SCALNYX) va s'interfacer avec Cingulata pour optimiser l'exécution du circuit booléen (format BLIF) et générer automatiquement un nouveau code parallèle de haute performance.

L'architecture de SCALGO est basée sur une philosophie de séparation des domaines entre le modèle de calcul, la dépendance de données, l'optimisation pour la performance, le calcul parallèle et son ordonnancement. Elle regroupe quatre composantes de calculs :

— Modèle de calcul : construction des langages dédiés à des problèmes algorithmiques (DSL : domain specific language), théories des graphes.
— Compilation : théorie de transformation de code source, analyse lexique et syntaxique, représentation intermédiaire, optimisation et génération de code.
— Programmation parallèle et distribuée : langage concurrent, développement des runtime parallèle, modèle de programmation, design patterns, protocole de communication.
— Architectures matérielles : maîtrise des architectures multi-cœur , many-cœur, GPUs, FPGA, vectorisation, accélérateurs exotiques, architectures serveurs et clouds, architecture mémoires, réseaux.

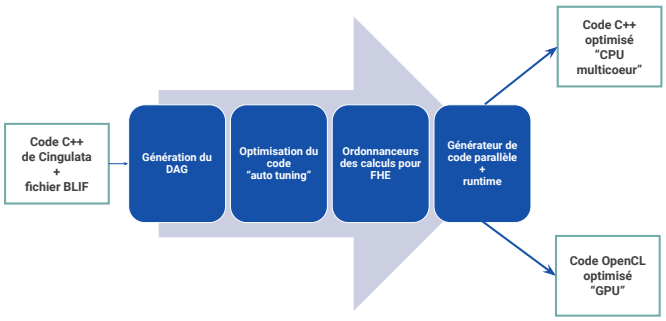

FIGURE 3. SCALGO : proposition de l'architecture.

La méthodologie de SCALGO est la suivante :

1) Entrée de Code original : Le code en entrée (typiquement écrit en C++) représente le code algorithmique en sortie du compilateur Cingulata avec le fichier du circuit booléens en format BLIF. Ce code est écrit en séquentiel sans parallélisme.
2) Génération du DAG : Les outils de compilation SCALGO génèrent une représentation de type DAG (directed acyclic graph), en décrivant les dépendances de données et noyaux, qui met en évidence les opportunités de parallélisme.
3) Optimisation : SCALGO présente de nombreuses options de paramètres d'optimisation pour permettre l'exploration de l'espace de conception. Par exemple, l'utilisateur peut choisir d'exploiter des localités spatiales (dans un seul noyau) et temporaires (dans deux ou plus noyaux topologiquement contigus), vectorisation, le parallélisme, l'équilibrage de charge, la prélecture, etc.
4) Ordonnancement des calculs : basé sur le graphe de dépendance de données et les paramètres d'optimisations, l'ordonnanceur de SCALGO trouvera un ordonnancement numériquement correcte entre les calculs des noyaux et identifiera la configuration optimale pour la performance. Il trouvera aussi des optimisations globales et locales qui sont impossibles à trouver par les développeurs ou les compilateurs classiques.
5) Générateur de code : finalement, SCALGO génère plusieurs configurations du système exécutés

par un runtime parallèle basé sur le modèle de programmation d'acteurs [51] ce qui permet de retrouver les optimisations plus efficaces.

VI. CONCLUSION

Le domaine du chiffrement homomorphe appliqué à l'intelligence artificielle en général, et au machine learning en particulier, n'est plus un rêve académique, mais s'approche de plus en plus des réalisations pratiques et on peut envisager avec confiance son industrialisation. Les industriels du monde entier ont besoin de l'aide de fournisseurs de Machine Learning pour réduire leurs coûts d'exploitation en entretenant et en optimisant leurs flux de processus. Cependant, ils ne le feront que s'ils sont assurés de la confidentialité de leurs données, même auprès de leurs fournisseurs de ces services. Jusqu'à présent, aucune solution n'existait pour répondre à ce besoin. Le cryptage entièrement homomorphe (FHE) changeait le jeu car il permettrait aux fournisseurs de travailler sur les données des clients sans jamais les déchiffrer et révéler le contenu original. Cette technologie protègerait le contenu des données pour des industriels, en fournissant des services analytiques sécurisés et fiables.

En conséquence, pour que la solution soit économiquement viable dans l'industrie, il reste à relever plusieurs verrous techniques :

- Architecture mémoire : chaque bit encrypté avec du chiffrement homomorphe pourra augmenter le nombre de données d'un facteur allant de 10^3 à 10^6 en fonction du niveau de sécurité requise. Du coup la technologie de mémoire (HBM, Optane, DDR4, SSD, cache) aura un impact très fort sur la performance. De plus, la façon d'organiser les données dans la hiérarchie mémoire et y accéder d'une façon intelligente et performante est primordiale.
- Méthode de chiffrement : le compilateur devra supporter d'autres techniques d'encryptage que BFV et TFHE pour adresser efficacement le plus grand nombre d'applications.
- Framework de développement dédié : ce framework permet de synthétiser un circuit booléen FHE optimisé pour une architecture cible à partir d'un algorithme machine learning. Il sert à l'évaluation de l'impact du FHE pour des fonctions machine learning et à l'automatisation de l'intégration du FHE pour différents cryptosystèmes. Il se basera sur des bibliothèques d'opérateur ML optimisés pour le FHE et pour plusieurs cibles hardware. L'objectif de ce framework est d'intégrer le chiffrement homomorphe dans les applications par des non-experts du métier de la cryptographie.
- Accélérateur dédié : un accélérateur permettant de répondre aux verrous technologiques et économiques, ainsi qu'à l'efficacité énergétique s'imposera. Les architectures hardware comme FPGA [52], GPU, ou Kalray MPPA [50] sont des pistes très sérieuses.

RÉFÉRENCES

[1] ResearchAndMarkets. Machine learning as a service (mlaas) - global market outlook (2017-2023). [Online]. Available : https://www.researchandmarkets.com/reports/4480635/machine-learning-as-a-service-mlaas-global

[2] CEA-LIST. Cingulata is a compiler toolchain and rte for running c++ programs over encrypted data by means of fully homomorphic encryption techniques. [Online]. Available : https://github.com/CEA-LIST/Cingulata

[3] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 32, no. 4, pp. 169–178, 1978.

[4] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptology—EUROCRYPT'99*, 1999, pp. 223–238.

[5] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[6] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pp. 309–325.

[7] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 868–886.

[8] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption." *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012.

[9] J.-C. Bajard, J. Eynard, A. Hasan, and V. Zucca, "A full rns variant of fv like somewhat homomorphic encryption schemes," Cryptology ePrint Archive, Report 2016/510, 2016.

[10] S. Halevi, Y. Polyakov, and V. Shoup, "An improved RNS variant of the BFV homomorphic encryption scheme," *IACR Cryptology ePrint Archive*, vol. 2018, p. 117, 2018. [Online]. Available : http://eprint.iacr.org/2018/117

[11] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors : Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 75–92.

[12] L. Ducas and D. Micciancio, "Fhew : bootstrapping homomorphic encryption in less than a second," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 617–640.

[13] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *Advances in Cryptology–ASIACRYPT 2010*. Springer, 2010, pp. 377–394.

[14] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe." in *ASIACRYPT (1)*, ser. Lecture Notes in Computer Science, T. Takagi and T. Peyrin, Eds., vol. 10624. Springer, 2017, pp. 377–408. [Online]. Available : http://dblp.uni-trier.de/db/conf/asiacrypt/asiacrypt2017-1.html#ChillottiGGI17

[15] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song, "Homomorphic encryption for arithmetic of approximate numbers." in *ASIACRYPT (1)*, ser. Lecture Notes in Computer Science, T. Takagi and T. Peyrin, Eds., vol. 10624. Springer, 2017, pp. 409–437. [Online]. Available : http://dblp.uni-trier.de/db/conf/asiacrypt/asiacrypt2017-1.html#CheonKKS17

[16] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for approximate homomorphic encryption." *IACR Cryptology ePrint Archive*, vol. 2018, p. 153, 2018. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2018.html#CheonHKKS18

[17] C. Boura, N. Gama, and M. Georgieva, "Chimera : a unified framework for b/fv, tfhe and heaan fully homomorphic encryption and predictions for deep learning." *IACR Cryptology ePrint Archive*, vol. 2018, p. 758, 2018. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2018.html#BouraGG18

[18] J. Cathébras, A. Carbon, P. Milder, R. Sirdey, and N. Ventroux, "Data flow oriented hardware design of rns-based polynomial multiplication for she acceleration." *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 69–88, 2018. [Online]. Available : http://dblp.uni-trier.de/db/journals/tches/tches2018.html#CathebrasCMSV18

[19] J. Cathébras, A. Carbon, R. Sirdey, and N. Ventroux, "An analysis of fv parameters impact towards its hardware acceleration." in *Financial Cryptography Workshops*, ser. Lecture Notes in Computer Science, M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. A. Ryan, V. Teague, A. Bracciali, M. Sala, F. Pintore, and M. Jakobsson, Eds., vol. 10323. Springer, 2017, pp. 91–106. [Online]. Available : http://dblp.uni-trier.de/db/conf/fc/fc2017w.html#CathebrasCSV17

[20] P. Méaux, C. Carlet, A. Journault, and F. Standaert, "Improved filter permutators : Combining symmetric encryption design, boolean functions, low complexity cryptography, and homomorphic encryption, for private delegation of computations," *IACR Cryptology ePrint Archive*, vol. 2019, p. 483, 2019. [Online]. Available : https://eprint.iacr.org/2019/483

[21] A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey, "Stream ciphers : A practical solution for efficient homomorphic-ciphertext compression," *J. Cryptology*, vol. 31, no. 3, pp. 885–916, 2018. [Online]. Available : https://doi.org/10.1007/s00145-017-9273-9

[22] P. Méaux, A. Journault, F.-X. Standaert, and C. Carlet, "Towards stream ciphers for efficient fhe with low-noise ciphertexts." *IACR Cryptology ePrint Archive*, vol. 2016, p. 254, 2016. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2016.html#MeauxJSC16

[23] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner, "Ciphers for mpc and fhe." *IACR Cryptology ePrint Archive*, vol. 2016, p. 687, 2016. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2016.html#AlbrechtRSTZ16

[24] S. Carpov, P. Dubrulle, and R. Sirdey, "Armadillo : A compilation chain for privacy preserving applications," in *Proceedings of the 3rd International Workshop on Security in Cloud Computing*, ser. SCC '15. New York, NY, USA : Association for Computing Machinery, 2015, p. 13–19. [Online]. Available : https://doi.org/10.1145/2732516.2732520

[25] S. Carpov, P. Aubry, and R. Sirdey, "A multi-start heuristic for multiplicative depth minimization of boolean circuits." *IACR Cryptology ePrint Archive*, vol. 2017, p. 483, 2017. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2017.html#CarpovAS17

[26] P. Aubry, S. Carpov, and R. Sirdey, "Faster homomorphic encryption is not enough : Improved heuristic for multiplicative depth minimization of boolean circuits," in *Topics in Cryptology - CT-RSA 2020 - The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24-28, 2020, Proceedings*, ser. Lecture Notes in Computer Science, S. Jarecki, Ed., vol. 12006. Springer, 2020, pp. 345–363. [Online]. Available : https://doi.org/10.1007/978-3-030-40186-3_15

[27] BERKELEY. Abc : A system for sequential synthesis and verification. [Online]. Available : https://people.eecs.berkeley.edu/~alanmi/abc/

[28] D. N. Kuate, S. Canard, and R. Sirdey, "Towards video compression in the encrypted domain : A case-study on the H264 and HEVC macroblock processing pipeline," in *Cryptology and Network Security - 17th International Conference, CANS 2018, Naples, Italy, September 30 - October 3, 2018, Proceedings*, ser. Lecture Notes in Computer Science, J. Camenisch and P. Papadimitratos, Eds., vol. 11124. Springer, 2018, pp. 109–129. [Online]. Available : https://doi.org/10.1007/978-3-030-00434-7_6

[29] K. Singh, R. Sirdey, and S. Carpov, "Practical personalized genomics in the encrypted domain." in *FMEC*. IEEE, 2018, pp. 139–146. [Online]. Available : http://dblp.uni-trier.de/db/conf/fmec/fmec2018.html#SinghSC18

[30] O. Stan, M.-H. Zayani, R. Sirdey, A. B. Hamida, A. F. Leite, and M. Mziou-Sallami, "A new crypto-classifier service for energy efficiency in smart cities." in *SMARTGREENS*, C. Klein, B. Donnellan, and M. Helfert, Eds. SciTePress, 2018, pp. 78–88. [Online]. Available : http://dblp.uni-trier.de/db/conf/smartgreens/smartgreens2018.html#StanZSHLM18

[31] K. Singh, R. Sirdey, F. Artiguenave, D. Cohen, and S. Carpov, "Towards confidentiality-strengthened personalized genomic medicine embedding homomorphic cryptography." in *ICISSP*, P. Mori, S. Furnell, and O. Camp, Eds. SciTePress, 2017, pp. 325–333. [Online]. Available : http://dblp.uni-trier.de/db/conf/icissp/icissp2017.html#SinghSACC17

[32] N. Bouzerna, R. Sirdey, O. Stan, T. H. Nguyen, and P. Wolf, "An architecture for practical confidentiality-strengthened face authentication embedding homomorphic cryptography." in *CloudCom*. IEEE Computer Society, 2016, pp. 399–406. [Online]. Available : http://dblp.uni-trier.de/db/conf/cloudcom/cloudcom2016.html#BouzernaSSNW16

[33] S. Carpov, T.-H. Nguyen, R. Sirdey, G. Costantino, and F. Martinelli, "Practical privacy-preserving medical diagnosis using homomorphic encryption." in *CLOUD*. IEEE Computer Society, 2016, pp. 593–599. [Online]. Available : http://dblp.uni-trier.de/db/conf/IEEEcloud/IEEEcloud2016.html#CarpovNSCM16

[34] O. Stan, S. Carpov, and R. Sirdey, "Dynamic execution of secure queries over homomorphic encrypted databases." in *SCC@AsiaCCS*, S. Zhong and A. C. Squicciarini,

Eds. ACM, 2016, pp. 51–58. [Online]. Available : http://dblp.uni-trier.de/db/conf/ccs/scc2016.html#StanCS16

[35] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets : Applying neural networks to encrypted data with high throughput and accuracy," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 201–210.

[36] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network." *IACR Cryptology ePrint Archive*, vol. 2017, p. 35, 2017. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2017.html#ChabanneWMMP17

[37] H. Chabanne, R. Lescuyer, J. Milgram, C. Morel, and E. Prouff, "Recognition over encrypted faces." in *MSPN*, ser. Lecture Notes in Computer Science, E. Renault, S. Boumerdassi, and S. Bouzefrane, Eds., vol. 11005. Springer, 2018, pp. 174–191. [Online]. Available : http://dblp.uni-trier.de/db/conf/mspn/mspn2018.html#ChabanneLMMP18

[38] S. Ioffe and C. Szegedy, "Batch normalization : Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available : http://arxiv.org/abs/1502.03167

[39] F. Bourse, M. Minelli, M. Minihold, and P. Paillier, "Fast homomorphic evaluation of deep discretized neural networks." in *CRYPTO (3)*, ser. Lecture Notes in Computer Science, H. Shacham and A. Boldyreva, Eds., vol. 10993. Springer, 2018, pp. 483–512. [Online]. Available : http://dblp.uni-trier.de/db/conf/crypto/crypto2018-3.html#BourseMMP18

[40] M. Izabachène, R. Sirdey, and M. Zuber, "Practical fully homomorphic encryption for fully masked neural networks." in *CANS*, ser. Lecture Notes in Computer Science, Y. Mu, R. H. Deng, and X. Huang, Eds., vol. 11829. Springer, 2019, pp. 24–36. [Online]. Available : http://dblp.uni-trier.de/db/conf/cans/cans2019.html#IzabacheneSZ19

[41] C. Boura, N. Gama, M. Georgieva, and D. Jetchev, "Simulating homomorphic evaluation of deep learning predictions," in *Cyber Security Cryptography and Machine Learning - Third International Symposium, CSCML 2019, Beer-Sheva, Israel, June 27-28, 2019, Proceedings*, ser. Lecture Notes in Computer Science, S. Dolev, D. Hendler, S. Lodha, and M. Yung, Eds., vol. 11527. Springer, 2019, pp. 212–230. [Online]. Available : https://doi.org/10.1007/978-3-030-20951-3_20

[42] C. Boura, N. Gama, and M. Georgieva, "Chimera : a unified framework for b/fv, TFHE and HEAAN fully homomorphic encryption and predictions for deep learning," *IACR Cryptology ePrint Archive*, vol. 2018, p. 758, 2018. [Online]. Available : https://eprint.iacr.org/2018/758

[43] M. Zuber, S. Carpov, and R. Sirdey, "Towards real-time hidden speaker recognition by means of fully homomorphic encryption." *IACR Cryptology ePrint Archive*, vol. 2019, p. 976, 2019. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2019.html#ZuberCS19

[44] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2 : Deep speaker recognition," *CoRR*, vol. abs/1806.05622, 2018. [Online]. Available : http://arxiv.org/abs/1806.05622

[45] J. H. Cheon, D. Kim, and J. H. Park, "Towards a practical clustering analysis over encrypted data." *IACR Cryptology ePrint Archive*, vol. 2019, p. 465, 2019. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2019.html#CheonKP19

[46] A. Jäschke and F. Armknecht, "Unsupervised machine learning on encrypted data." *IACR Cryptology ePrint Archive*, vol. 2018, p. 411, 2018. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2018.html#JaschkeA18

[47] S. Carpov, N. Gama, M. Georgieva, and J. R. Troncoso-Pastoriza, "Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption." *IACR Cryptology ePrint Archive*, vol. 2019, p. 101, 2019. [Online]. Available : http://dblp.uni-trier.de/db/journals/iacr/iacr2019.html#CarpovGGT19

[48] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure logistic regression based on homomorphic encryption : Design and evaluation," *JMIR Med Inform*, vol. 6, no. 2, p. e19, Apr 2018. [Online]. Available : https://doi.org/10.2196/medinform.8805

[49] Nucypher. A gpu implementation of fully homomorphic encryption on torus. [Online]. Available : https://github.com/nucypher/nufhe

[50] Kalray. Kalray mppa manycore architecture. [Online]. Available : https://www.kalrayinc.com/

[51] SCALNYX. Simplx : C++ development framework for building reliable cache-friendly distributed and concurrent multicore software based on the actor model principle. [Online]. Available : https://github.com/scalnyx/simplx

[52] J. Cathébras, A. Carbon, P. Milder, R. Sirdey, and N. Ventroux, "Data flow oriented hardware design of rns-based polynomial multiplication for SHE acceleration," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 69–88, 2018. [Online]. Available : https://doi.org/10.13154/tches.v2018.i3.69-88