Combining differential privacy and homomorphic encryption for privacy-preserving collaborative machine learning

Arnaud Grivet Sébert

université PARIS-SACLAY

Combining differential privacy and homomorphic encryption for privacy-preserving collaborative machine learning

Approches combinant confidentialité différentielle et chiffrement homomorphe pour la protection des données en apprentissage automatique collaboratif

Thèse de doctorat de l'université Paris-Saclay

École doctorale 580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Informatique mathématique
Graduate School: Informatique et sciences du numérique. Référent: Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Institut LIST (Université Paris-Saclay, CEA)**, sous la direction de **Renaud Sirdey**, directeur de recherche, le co-encadrement de **Cédric Gouy-Pailler**, ingénieur de recherche

Thèse soutenue à Paris-Saclay, le 12 juin 2023, par

Arnaud GRIVET SÉBERT

Composition du jury

Membres du jury avec voix délibérative

David POINTCHEVAL Directeur de recherche, CNRS, ENS-PSL - Université Paris Sciences et lettres, France	Président
Melek ÖNEN Maîtresse de conférence (HDR), EURECOM - EDITE, Paris Sorbonne Université, France	Rapporteure & Examinatrice
Jan RAMON Directeur de recherche, INRIA - Université de Lille, France	Rapporteur & Examinateur
Rachid GUERRAOUI Professeur, EPFL, Suisse	Examinateur

THESE DE DOCTORAT

NNT : 2023UPASG037

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Approches combinant confidentialité différentielle et chiffrement homomorphe pour la protection des données en apprentissage automatique collaboratif

Mots clés: apprentissage automatique confidentiel, confidentialité différentielle, chiffrement homomorphe, apprentissage collaboratif

Résumé: L'objet de cette thèse est la conception de protocoles pour l'entraînement de modèles d'apprentissage automatique avec protection des données d'entraînement. Pour ce faire, nous nous sommes concentrés sur deux outils de confidentialité, la confidentialité différentielle et le chiffrement homomorphe. Alors que la confidentialité différentielle permet de fournir un modèle fonctionnel protégé des attaques sur la confidentialité par les utilisateurs finaux, le chiffrement homomorphe permet d'utiliser un serveur comme intermédiaire totalement aveugle entre les propriétaires des données, qui fournit des ressources de calcul sans aucun accès aux informations en clair. Cependant, ces deux techniques sont de natures totalement différentes et impliquent toutes deux leurs propres contraintes qui peuvent interférer : la confidentialité différentielle nécessite généralement l'utilisation d'un bruit continu et non borné, tandis que le chiffrement homomorphe ne peut traiter que des nombres encodés avec un nombre limité de bits. Les travaux présentés visent à faire fonctionner ensemble ces deux outils de confidentialité en gérant leurs interférences et même en les exploitant afin que les deux techniques puissent bénéficier l'une de l'autre.

Dans notre premier travail, SPEED, nous étendons le modèle de menace du protocole PATE (Private Aggregation of Teacher Ensembles) au cas d'un serveur honnête mais curieux en protégeant les calculs du serveur par une couche homomorphe. Nous définissons soigneusement quelles opérations sont effectuées homomorphiquement pour faire le moins de calculs possible dans le domaine chiffré très coûteux tout en révélant suffisamment peu d'informations en clair pour être facilement protégé par la confidentialité différentielle. Ce compromis nous contraint à réaliser une opération argmax dans le domaine chiffré, qui, même si elle est raisonnable, reste coûteuse. C'est pourquoi nous proposons SHIELD dans une autre contribution, un opérateur argmax volontairement imprécis, à la fois pour satisfaire la confidentialité différentielle et alléger le calcul homomorphe. La dernière contribution présentée combine la confidentialité différentielle et le chiffrement homomorphe pour sécuriser un protocole d'apprentissage fédéré. Le principal défi de cette combinaison provient de la discrétisation nécessaire du bruit induite par le chiffrement, qui complique l'analyse des garanties de confidentialité différentielle et justifie la conception et l'utilisation d'un nouvel opérateur de quantification qui commute avec l'agrégation.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Combining differential privacy and homomorphic encryption for privacy-preserving collaborative machine learning

Keywords: privacy-preserving machine learning, differential privacy, homomorphic encryption, collaborative learning

Abstract: The purpose of this PhD is to design protocols to collaboratively train machine learning models while keeping the training data private. To do so, we focused on two privacy tools, namely differential privacy and homomorphic encryption. While differential privacy enables to deliver a functional model immune to attacks on the training data privacy by end-users, homomorphic encryption allows to make use of a server as a totally blind intermediary between the data owners, that provides computational resource without any access to clear information. Yet, these two techniques are of totally different natures and both entail their own constraints that may interfere: differential privacy generally requires the use of continuous and unbounded noise whereas homomorphic encryption can only deal with numbers encoded with a quite limited number of bits. The presented contributions make these two privacy tools work together by coping with their interferences and even leveraging them so that the two techniques may benefit from each other.

In our first work, SPEED, we built on Private Aggregation of Teacher Ensembles (PATE) framework and extend the threat model to deal with an honest-but-curious server by covering the server computations with a homomorphic layer. We carefully define which operations are realised homomorphically to make as less computation as possible in the costly encrypted domain while revealing little enough information in clear to be easily protected by differential privacy. This trade-off forced us to realise an argmax operation in the encrypted domain, which, even if reasonable, remained expensive. That is why we propose SHIELD in another contribution, an argmax operator made inaccurate on purpose, both to satisfy differential privacy and lighten the homomorphic computation. The last presented contribution combines differential privacy and homomorphic encryption to secure a federated learning protocol. The main challenge of this combination comes from the fact that the encryption induces a quantisation of the noise, that complicates the differential privacy analysis and justifies the design and use of a novel quantisation operator that commutes with the aggregation.

Acknowledgements

This thesis has been a three-year journey, very enriching on the scientific point of view but also truly fulfilling human-wise: I met a lot of great people and made new friends.

This adventure originated thanks to the collaboration of two researchers, each one with his own speciality: Renaud the cryptographer and Cédric, the expert of machine learning. Together, they wanted to explore the combination of homomorphic encryption with differential privacy in machine learning. Before I even started my PhD, they had paved the way on this idea and this enabled me to start *in medias res*, straight into the "gory details", as Renaud would say, saving a precious time at the beginning of my PhD. My first thanks obviously go to them, for allowing me as a PhD candidate, and accompanying me during these three years. Thank you Renaud for your great ideas that solved a number of technical issues, for your enthusiasm regarding our work that really encouraged me. Thank you Cédric for your help with the code and the experiments (God knows I needed it), especially at the beginning with SPEED, for prompting me to finish what I had started, proposing deadlines when I was stubbornly stuck on a technical issue.

I would also like to thank Melek Önen, Jan Ramon, Rachid Guerraoui and David Pointcheval for being part of my defense's jury. I especially thank Jan Ramon, who was also part of my mid-term commitee and with whom we had very interesting discussions, both online and when he came to visit us in CEA; Jan really challenged our work and gave us enlightening insights to improve it.

Research work would be much less fruitful if it were a solitary process but, even apart from any productivity concern, as a non-self-contained individual, I could not have worked happily without the collaboration of Renaud and Cédric of course, Martin, Oana, Rafaël, Pierre-Emmanuel, Aymen, Aurélien, Abbass and Marina. I am particularly grateful to Martin and Oana who have been working with me during most of my PhD, and still are, for the many helpful discussions on the intricacies of advanced fully homomorphic encryption optimisations as well as for the fully homomorphic encryption implementation works which allowed to obtain the performance results for the contributions presented in the second part of this manuscript. Thank you also Martin for your permanent enthusiasm, being for SHIELD algorithm or for climbing sessions!

After these years in CEA, I met more than colleagues, I met friends. I would like to thank Fabiola for the moments and discussions we shared, for bearing me as a (bad) climbing teacher, for introducing me to arepas. Thank you Eduardo for the Portuguese-French bilingual conversations and *por matar saudades do Brasil aqui*. Thank you Jobic for the parties and the math discussions, Antoine for the many climbing sessions and talks about matrices, kurtosis or information theory. Thank you Baudouin for the fun we had in 2041, Elouan and Victor for the blackboard sessions. Thank you Etienne, always thoughtful and ready to joke, still waiting for climbing with Lucas! Thank you Ilyes for the philosophical conversations. Thank you Amal for the Arabic lessons and the improvised breaks. Thank you Camilo el salsero and Florian the Farsi expert for the etymology club that died young but should rise back from the ashes. Thank you Lucie for our talks about Japan, Antonin for the meals with Eduardo after our lessons in Collège de France. Thank you Edwin for the football sessions, and all the others from the lab.

I actually did not spend three years in CEA but four and a half, since I had worked before my PhD with Jean-Philippe in C-BORD project. When I started this short-term contract, I was only curious about research and did not know it would be a long-term orientation. I am grateful to Jean-Philippe for introducing me to research and to CEA, for allowing me to write and publish my first papers and travel to so many

places, in particular back to Rio that I wanted to see again. Those times were also very good times in CEA, where I made long-lasting friendships. Let me mention those old-times people, aka the midday eaters, since this period also taught me a lot about research and contributed to my education towards PhD. Sandra for the Spanish lessons, the chestnuts and the many CEA PhD students I met thanks to her, Ismaïl for the hikes, the coding support, Vincent for the rider spirit, Régis my travel partner to Spain and Brazil, Shivani for offering me new options to diversify my "vegan" diet, Andrey the funniest of the fathers. Baptiste for the many trips to Fontainebleau and for Mimizaaang, Hung for his evergreen happy nature, Oudom the only one beating me at eating. I also thank Rafaël for the advice and for starting the work with his own PhD.

Between my two contracts in CEA, I had the opportunity to work on a totally different subject, computational social choice, in LIP6. I really enjoyed working on this and I am grateful to Patrice, Paolo and Nicolas for giving me this chance, also to Jérôme Lang for introducing me to them. I also thank the LIP6 artists for welcoming me and for the Buet: Gaspard who would come every day to the other corridor to take me to lunch, Adèle, Kostas, David, Anne-Elisabeth, Marvin, Franco, Ismaïl again.

Let me also thank all my friends, from high school, prépa, engineering school, Brazil for supporting me and making life so cheerful and exciting. I will not mention their names here but I think about each of them. Thanks to my former and current flatmates from Montrouge, it was really refreshing to come back home and chill with you guys after a day of work.

Last but not least, I will never be able to thank my family enough, especially my parents. They enabled me to choose the life I want to live by the raising and education they provided me, supported me, selflessly giving all they can for me and my sister. Saying that I would be nothing without them is an obvious tautology but is still worth saying. I thank my sister Anne-Clotilde for all the years we spent together, making all my young years much more fun, and I thank my brother-in-law Antoine, for whom I realise the mention "in-law" is getting superfluous.

À mes parents.

Synthèse en français

L'objet de cette thèse est la conception de protocoles pour l'entraînement collaboratif de modèles d'apprentissage automatique avec protection des données d'entraînement. Pour ce faire, nous nous sommes concentrés sur deux outils de confidentialité, la confidentialité différentielle et le chiffrement homomorphe. La confidentialité différentielle utilise du bruit aléatoire pour cacher l'information sensible des individus. Grâce à celle-ci, il est possible de fournir un modèle fonctionnel protégé des éventuelles attaques sur la confidentialité par les utilisateurs finaux. Le chiffrement homomorphe, quant à lui, désigne un ensemble de techniques cryptographiques qui permettent de réaliser des opérations sur des données chiffrées. En employant ce type de chiffrement, on peut utiliser un serveur comme intermédiaire totalement aveugle entre les propriétaires des données, qui fournit des ressources de calcul sans aucun accès aux informations en clair. Cependant, ces deux outils sont de natures totalement différentes et impliquent tous deux leurs propres contraintes qui peuvent interférer : la confidentialité différentielle nécessite généralement l'utilisation d'un bruit continu et non borné, tandis que le chiffrement homomorphe ne peut traiter que des nombres encodés avec un nombre limité de bits. Les travaux présentés visent à faire fonctionner ensemble ces deux outils de confidentialité en composant avec leurs interférences et même en les exploitant afin que les deux techniques puissent bénéficier l'une de l'autre.

Dans notre premier travail SPEED (*Secure, PrivatE, and Efficient Deep learning*, soit apprentissage profond sécurisé, confidentiel et rapide), une base de données publique est étiquetée en agrégeant le savoir de plusieurs modèles professeurs, via un serveur qui choisit la réponse la plus fréquente parmi les réponses des professeurs, qui sont vues comme des votes. La base de données publique, ainsi étiquetée, est utilisée pour entraîner un modèle étudiant. SPEED est inspiré du protocole PATE (*Private Aggregation of Teacher Ensembles*, soit agrégation confidentielle d'ensembles de professeurs) mais, contrairement au modèle de menaces de celui-ci, le serveur est considéré comme une menace pour la confidentialité des données des professeurs. Les calculs du serveur sont donc protégés par une couche homomorphe. Nous définissons soigneusement quelles opérations sont effectuées homomorphiquement pour faire le moins de calculs possible dans le domaine chiffré, très coûteux, tout en révélant suffisamment peu d'informations en clair pour que les données soient facilement protégées par la confidentialité différentielle. Ce compromis nous contraint à réaliser une opération argmax dans le domaine chiffré, qui, même si elle est raisonnable, reste coûteuse.

C'est pourquoi, dans une autre contribution, nous proposons SHIELD (*Secure and Homomorphic Imperfect Election via Lightweight Design*, soit élection imparfaite sécurisée et homomorphe de conception algorithmique légère), un opérateur de sélection du vote le plus fréquent volontairement imprécis, à la fois pour satisfaire la confidentialité différentielle et alléger le calcul homomorphe. Il est particulièrement approprié au protocole SPEED mais son champ d'application se veut plus général puisqu'il pourrait être utilisé pour des élections avec garanties de confidentialité. La sortie de l'opérateur SHIELD est obtenue en tirant aléatoirement parmi les votes et l'opérateur est ainsi intrinsèquement probabiliste. Un résultat crucial de cette contribution est que ce comportement probabiliste donne à SHIELD des garanties de confidentialité différentielle sans nécessiter une addition de bruit aléatoire. Pour améliorer la précision, l'entropie de la distribution des votes est réduite grâce à un nombre paramétrable d'additions et de multipilcations homomorphes, moins coûteuses à réaliser qu'un argmax homomorphe exact. Par construction, cet opérateur dépasse les compromis classiques de la cryptographie et de la confidentialité différentielle, respectivement

sécurité-performance et confidentialité-précision, en alignant les objectifs de performance et de confidentialité : relâcher les paramètres permet des calculs efficaces tout en offrant "gratuitement" des garanties de confidentialité.

La dernière contribution présentée combine la confidentialité différentielle et le chiffrement homomorphe pour sécuriser un protocole d'apprentissage fédéré. Le principal défi de cette combinaison provient de la discrétisation nécessaire du bruit induite par le chiffrement. En effet, pour garder des garanties de confidentialité différentielle du point de vue du serveur, les clients se chargent eux-mêmes de bruiter leurs mises à jour avant de les envoyer au serveur, de sorte que le bruit total après agrégation fournisse les garanties de confidentialité requises. Une discrétisation classique induit un bruit agrégé suivant une distribution complexe ne permettant pas une analyse de confidentialité simple. Ceci justifie la conception et l'utilisation d'un nouvel opérateur de discrétisation, fondé sur la distribution de Poisson, et qui commute avec l'agrégation. Étant ainsi équivalente à un post-traitement, la discrétisation n'a aucun impact sur les garanties de confidentialité qui s'avèrent donc être les mêmes que celles du très classique mécanisme gaussien.

Contents

List of Figures

List of Tables

"Sa sacrée Majesté le Hasard fait les trois quarts de la besogne..."
Frederick II the Great, king of Prussia

Introduction

Machine Learning (ML) techniques have become ubiquitous in almost all fields of industry and our daily lives. The used algorithms, and especially neural networks, the most popular ones in many applications among which image processing and natural language processing, require massive amounts of data to get trained and reach the impressive accuracy that made their success. This huge need for data together with the omnipresence of the Internet and the boom of communication exchanges made data the "new oil" (Clive Humby, 2006).

In parallel, confidentiality has raised greater and greater concerns as evidenced by the new regulations on data privacy (*e.g.* GDPR [1] in the EU). The COVID crisis has entailed an increase of personal data sharing and people tracking with the goal of limiting the spread of the virus. In reaction, the interest and questionings about privacy have become more and more vivid. While all kinds of data might be considered as private, some fields like healthcare (*e.g.* HIPAA [40] in the USA), finance, commercial strategy are especially sensitive to privacy breaches. In this context, it is well known that, once trained, a machine learning model may indirectly release some information about its training data [87]. Many researchers have exhibited attacks on machine learning models, assuming that the adversary, e.g. an end-user of the model, has access to the parameters of the model (white-box access) or even only to the output of the inference on some queries it made to the model (black-box access) [73, 158, 167]. These attacks are even more likely with the emergence of machine learning as a service [151] in recent years: an adversary, by playing the role of a customer, can have access to many trained models and learn information about their training data along the queries.

A general trend of the recent advances in machine learning is to train a model using knowledge from several independent entities that take part in the training process. We call this kind of learning framework *collaborative learning*. Besides the end-users, other potential adversaries appear in the context of collaborative learning. Indeed, a learning framework involving several data owners usually resorts to a server that gathers and computes information from the data owners. This information access may jeopardise the training data privacy if the server is not to be trusted. Furthermore, federated learning, a collaborative learning paradigm introduced in 2016 by McMahan et al. [126] to protect the data by keeping them on the owner's device, actually brings in a new kind of adversary - the data owners themselves, known as clients. Indeed, at each round of the iterative learning process, the clients get some information derived from all the training data, thus giving any client the opportunity to retrieve information from the other clients.

To deal with these issues, one of the most popular approaches to data privacy is Differential Privacy (DP), introduced by Dwork et al. [63], which quantifies the amount of information leaked by the output of a mechanism about its input. This notion of privacy implies that the considered mechanism is probabilistic, which is often achieved by applying carefully parameterised random noise to a deterministic mechanism. Since 2006, DP has been actively studied in the context of machine learning (see *e.g.* [102]).

Cryptography is another field that helps providing protection to sensitive data, in particular

at training time, especially since the recent emergence of techniques for computing directly over encrypted data like Homomorphic Encryption (HE) [149,189] and multi-party computation [134] (and in particular secure aggregation [25]). As such, these techniques allow a server to perform secure, blind computations without access to the inputs, and, in the case of HE, to the outputs either. While this thesis focuses on the threats coming from honest-but-curious adversaries, i.e. adversaries that properly execute their tasks, HE can also be associated in some cases to tools such as verifiable computing to bring additional computation integrity guarantees, including in the context of federated learning [122].

The revolution of AI and especially deep learning that poses great challenges in terms of privacy together with the discovery of novel tools in cryptography and statistics to enhance data privacy give rise to an exciting new research field. The interest for privacy-preserving machine learning boosts research efforts about privacy-enhancing techniques while allowing AI applications to flourish in the respect of people's right to privacy.

Summary of the thesis: The purpose of this PhD is to design protocols to collaboratively train machine learning models while keeping the training data private. To do so, we focused on two privacy tools, namely DP and HE. While DP enables to deliver a functional model immune to attacks on the training data privacy by end-users, HE allows to make use of a server as a totally blind intermediary between the data owners, that provides computational resources without any access to clear information. Yet, these two techniques are of totally different natures and both entail their own constraints that may interfere: for example, DP generally requires the use of continuous and unbounded noise whereas HE can only deal with numbers encoded with a quite limited number of bits. The presented contributions make these two privacy tools work together by coping with their interferences and even leveraging them so that the two techniques may benefit from each other.

In our first work, SPEED, for Secure, PrivatE, and Efficient Deep learning, a public unlabelled database is labelled by aggregating the knowledge of several teacher models, via a server that applies the plurality rule on the teachers' answers, seen as votes. The public database, once labelled, is used to train a student model. Both the end-users of the student model and the server are deemed honest-but-curious adversaries. We cover the server computations by a homomorphic layer and carefully define which operations are realised homomorphically to make as less computations as possible in the computationally costly homomorphic domain while revealing little enough information in clear to be easily protected by DP. In fact, revealing the histogram of the teachers' votes in clear would have required to apply too much noise to maintain reasonable DP guarantees, that is why the argmax operation is realised in the encrypted domain.

In another contribution, we propose SHIELD, Secure and Homomorphic Imperfect Election via Lightweight Design, an argmax operator which is made inaccurate on purpose, both to satisfy DP and lighten the homomorphic computations. It is particularly appropriate to SPEED framework but may have more general applications. SHIELD obtains its output by sampling in the teachers' votes distribution and, as such, is inherently probabilistic. A crucial result of this contribution is that this probabilistic behaviour makes SHIELD differential private, without need for noise addition. To increase the accuracy, the entropy of the teachers' votes distri-

bution is lowered thanks to a limited and parameterisable number of homomorphic additions and multiplications, simpler to compute than an exact homomorphic argmax. By construction, this operator goes beyond the classical trade-offs in cryptography and DP, respectively security-performance and privacy-utility, by aligning the performance and privacy objectives: loose parameters that allows for efficient computations also offer privacy "for free".

The last contribution of this manuscript combines DP and HE to secure a federated learning protocol. The main challenge of this combination comes from the necessary quantisation of the noise induced by encryption. Indeed, to keep the training differentially private from the point of view of the server, the clients are in charge of noising their updates before sending them to the server, so that the total noise after aggregation ensures the required DP guarantees. Classical quantisation results in a complex aggregated noise that does not allow for a simple DP analysis. This justifies the design and use of a novel stochastic quantisation operator, based on the Poisson distribution, that commutes with the aggregation and, being equivalent to a post-processing, does not have any influence on the DP analysis that boils down to the vanilla Gaussian mechanism analysis.

Outline of the manuscript: In the first part of this manuscript, we introduce the context and state of the art, along with some technical background. In a first chapter, we reflect on the possible reasons of the importance of data privacy in our society, giving examples of privacy breaches that would clearly need to be addressed, according to a large consensus. We also present some of the mainstream regulations about data privacy in the world, most of them being quite recent, before detailing the terminology about privacy concerns in the different relevant fields. After summarising the notations that will be employed throughout the manuscript, Chapter 2 presents the technical tools usually used in privacy-preserving machine learning. We introduce DP, mixing technical definitions and properties with intuitive insight, along with toy examples, research works and real-life use-cases. A second section deals with cryptographic primitives and most importantly HE and its several flavours. We briefly mention some works of the literature. Finally, we compare cryptographic security and DP, seen as different levels of the privacy-utility trade-off. In the last chapter of this part, we introduce federated learning and especially the federated averaging algorithm before reviewing the literature about collaborative learning protocols that protect the training data privacy, most of the works involving federated learning as it is currently the most popular collaborative learning framework. We focus on protocols that combine DP and cryptographic primitives to get the best of both worlds.

In a second part, we thoroughly present our three main contributions, named after the corresponding articles that we summarised above: *SPEED: Secure, PrivatE, and Efficient Deep learning*, *When approximate design for fast homomorphic computation provides differential privacy guarantees* and *Combining homomorphic encryption and differential privacy in federated learning*.

The manuscript is concluded by an overview of how our contributions can help in addressing the issues presented in the first part. We also propose perspectives on applications of our works as well as more theoretical lines of research.

Publications and talks

Articles in scientific journals and conferences:

- Grivet Sébert, A., Pinot, R., Zuber, M., Gouy-Pailler, C., Sirdey, R. (2021). **SPEED: secure, PrivatE, and efficient deep learning**. *Machine Learning*, 110(4), 675-694 (reference [83]), presented at ECML-PKDD 2020
- Madi, A., Stan, O., Mayoue, A., Grivet-Sébert, A., Gouy-Pailler, C., Sirdey, R. (2021). **A secure federated learning framework using homomorphic encryption and verifiable computing**. In *2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS)* (pp. 1-8). IEEE (reference [122])
- Grivet Sébert, A., Sirdey, R., Stan, O., Gouy-Pailler, C. (2022). **Protecting Data from all Parties: Combining FHE and DP in Federated Learning**. *arXiv preprint arXiv:2205.04330* (reference [84]) (to be submitted)
- Grivet Sébert, A., Zuber, M., Sirdey, R., Stan, O., Gouy-Pailler, C. (2023). **When approximate design for fast homomorphic computation provides differential privacy guarantees**. *arXiv preprint arXiv:2304.02959* (reference [85]) (to be submitted)

Article in scientific popularisation journal:

- Sirdey, R., Grivet Sébert, A., Gouy-Pailler, C. (2022). **[Cahier technique] Cryptographie homomorphe: l'art de partager sans divulguer** (in French). *Industrie et technologies* (reference [168])

Invited talks:

- Talk about **Protecting Data from all Parties: Combining FHE and DP in Federated Learning** at Paris Privacy Preserving AI Meetup, June, 8th, 2022
- **Machine learning without jeopardising the training data** at Principles of Distributed Learning (PODL) workshop in ACM Principles of Distributed Computing (PODC) 2022, Salerno, Italy, July, 25th, 2022

Part I

Context and state of the art

1 - On the importance of data privacy

In 1889, the emergence of photography and sensational journalism urged attorney Samuel Warren and future U.S. Supreme Court Justice Louis Brandeis to write an article about the right to privacy, defined as "the right to be let alone" [183]. 117 years later, while machine learning was about to explode, mathematician and entrepreneur Clive Humby pronounced his famous sentence "data is the new oil". The following years confirmed Humby's quote. Nowadays, the GAFAM, whose market capitalisation is of the order of magnitude of a trillion dollars, have access to tremendous amounts of data about their users - habits, decision-making, preferences. They can use these data to improve their recommendation systems, their search engines, target the advertisements, turning them to always be more attractive, more and more used and then giving them access to more and more data. But this virtuous circle and the eagerness for personal data it entails is dangerous for people's privacy.

1.1 . Why must some information remain private?

Our society greatly values privacy. Is there an ethical, moral reason that justifies this care about privacy? Far from claiming to exhaustively study the difficult philosophical question of privacy, we rather ask some questions and present some very basic ideas to raise the awareness on an issue which is not as obvious as one may think *a priori*. It seems indeed quite reasonable to ask these questions before extensively discussing technical ways of ensuring privacy throughout this manuscript.

1.1.1 . Individuals' privacy

First of all, let us clarify what is meant by privacy. In this chapter, and in general in this thesis, the term privacy will refer to the control over information about oneself and over whom or what can have access to it. This being stated, we may now wonder why we care about privacy.

The most obvious, and perhaps more relevant in practise, answer comes from the *utilitarian*[1] point of view: some information may be used maliciously. For instance, one may want to hide some goods or the fact that one owns these goods for fear from being stolen, as well as keep its bank password secret. More specifically to the Internet era, cookies that track an individual's activity on the web might be used to increase the price of goods or services the individual is interested in (plane tickets is a famous example). One may also want to keep its location private to prevent an aggression. In this view, privacy is useful to prevent a danger or a prejudice.

Fairness can also be a justification for privacy. The fight against discrimination of any kind may need the hiding of some information when, unfortunately, someone in charge of judging a person or a person's work is suspected from being biased (this threat justifies anonymised exams, blind reviews for scientific papers).

[1]Utilitarianism is a doctrine that states that something's value must be measured by its usefulness.

More generally, it seems that human beings are often reluctant to share personal information because of a vague distrust towards whom could get this information, not necessarily knowing what could happen in case of revelation, as if the revelation of private information made us more exposed to harm. The physical counterpart of this distrust includes a certain coyness to show nudity and, also, information regarding health status, another exemplar application of privacy techniques. Indeed, knowing that somebody is ill is knowing this person is vulnerable. Hence, considering medical data as sensitive might be an heritage of the fear of showing its vulnerability, which could constitute an incentive to aggression[2].

In fact, knowing one's weaknesses, or, in general, having a detailed vision of one's personality, gives a lot of power: personal data can be used by an adversary to manipulate people. Indeed, if one knows somebody's political orientations or religious beliefs, it will be easier to convince him/her to have a particular behaviour. The scandal of Cambridge Analytica illustrates this risk quite well. The consulting company Cambridge Analytica designed a personality quiz and prompted people to answer it by offering money and pretending it was for scientific purposes. Once someone had answered the quiz, she had to access her Facebook profile to get the financial reward. Cambridge Analytica then had access to the general information and the likes from her Facebook profile and her Facebook friends' profiles. The company also bought, sometimes illegally, additional information about its victims (subscriptions to magazines, travel tickets). Thanks to this information, Cambridge Analytica sent targeted messages via social networks to the most persuadable of its victims in order to manipulate them. Among the clients of Cambridge Analytica were republican supporters during the US presidential election of 2016 when Cambridge Analytica's messages influenced people so that either they vote for Donald Trump, or, if they were deemed too hard to convince, they do not vote for the election.

Privacy also allows an individual to act without fearing any judgement, influence, threat or retaliation, which is the rationale for vote secrecy in democracy for example. This leads us to a more abstract view of privacy: some authors [59, 162] make the hypothesis that we value privacy because it allows us to act more freely, in a kind of bubble protected from the interference and influence of other people. Of course, this argument can be reversed: without being seen by anybody, one might do any bad action with impunity, as suggests Plato's myth of the ring of Gyges. Nevertheless, it seems reasonable to think that some privacy is necessary to keep individuals' independence and identity. Without it, the least action would be constrained by social norms and we may argue that this would greatly harm creativity: an idea that would be considered as shameful, taboo or simply weird might evolve in a disruptive innovation. Westin [184] also argues that social life is stressful and we need moments to rest from this pressure and the urge of social role-playing. Actually, people also need privacy to keep this very social role-playing intact by hiding their flaws: people always want to appear better than they are.

For whom does not want to bother with or is not convinced by a reflection about the possible intrinsic value of privacy, it may simply be argued that privacy inherits its value from the value of information. Prosaically, it is for the very fact that information is of great economic

[2]A more trivial argument is that being aware of a client's illness makes insurance companies increase their price for this client.

value nowadays that it should be protected from curious ears: "if people can make money out of my data, I want my piece of the cake and I will keep my data private until someone buys me the right to access them, at due price".

1.1.2 . Non-individual confidentiality

Whereas *privacy* refers to limiting the access of information about an individual, *confidentiality* amounts to the non-divulgation of information in a more general context, possibly beyond the individual sphere. In the military domain, one of the important application fields of privacy in computer science, the confidentiality of a state is involved and the disclosure of, say, technologies or strategic moves may endanger the sovereignty of a country, and, unfortunately, human lives. A famous illustration of the crucial importance of military confidentiality is the breaking of the German cryptographic system Enigma by Alan Turing during WWII: some experts estimated that this cryptanalysis shortened the war by two years. While this privacy breach helped to shorten the war, one easily imagines the terrible consequences of strategic information leakage in other circumstances. Confidentiality is also crucial in any context that involves competition like, for example, industry and the fight against industrial espionage, based on companies' confidentiality.

The techniques explored in this manuscript may help to protect individuals' privacy or confidentiality of bigger entities, depending on the application but we will invariably use the word *privacy* to refer to the non-disclosure of sensitive information.

1.2 . Regulations on data privacy

In the European Union, the General Data Protection Regulation (GDPR) gives a unified legal framework for personal data collection and processing in the European Union since 2018. *Personal data* is defined as any information concerning an identified or identifiable physical person. GDPR is based on several principles:

- *finality and minimisation*: any data collected must be needed for a specific purpose and any data that is not required by a purpose must not be collected
- *transparency*: individuals must be informed of which of their data are revealed, of the use that is made of these data and of their rights concerning their data
- a company that has access to an individual's personal data must facilitate the access, modification or removal by the individual
- personal data must be deleted, anonymised or archived after a clearly defined amount of time
- *privacy by design*: a company having access to personal data must take all useful measures to guarantee the security of these data, during all the life cycle of the data, from collection to deletion
- *privacy by default*: by default, the highest level of privacy must be enforced

With the adoption of GDPR, Europe has kickstarted a wave of data privacy regulations around the world. Indeed, to be able to keep on trading with EU a country now requires to be GDPR-compliant. A few countries outside Europe are considered as such by the European Commission, among which Canada, Japan, New-Zealand, Argentina.

In the USA, there is no privacy regulation on the federal level but the states have their own privacy laws, like the California Consumer Privacy Act (CCPA) which is quite similar to GDPR. Still in the USA, the Health Insurance Portability and Accountability Act (HIPAA), which was actually promulgated 22 years before GDPR, made the Department of Health and Human Services (HHS) promulgate five rules to regulate the exchange of individually identifiable health information between the actors of the medical field.

These regulations, and in particular the two last aforementioned GDPR principles, justify the use of privacy tools and privacy-preserving protocols like the ones studied in this thesis. Yet, one might also imagine that the increasing practicality of such tools will enable data trading with even more privacy constraints than the ones stated in the current laws, for example compelling servers to process only encrypted data as HE would allow. In addition to the law-to-technology push, we may thus observe in the future a technology-to-law push, making data exchange more private than it would have been thought possible.

1.3 . Deanonymisation: Breach privacy can be easier than you think

Anonymisation is the fact of removing from the columns of a database the information that may (obviously) allow to identify an individual, such as name, address, social security number. Several events have shown the weakness of this concept to protect people's privacy.

In [138], Ohm recalls how, as early as the mid-1990s, Latanya Sweeney theatrically invalidated the anonymisation guarantees. In Massachusetts, USA, the Group Insurance Commission had released anonymised medical records of state employees. William Weld, Governor of Massachusetts, claimed that the employees' privacy was guaranteed by the removal of identifiers. Thanks to the combination of ZIP code, birth date and sex data, Sweeney was able to identify patients from the GIC data, including Weld, to whom she sent his own health records at his office.

Another example, quite famous, is the Netflix Prize scandal. In 2006, Netflix released the ratings and date of ratings of nearly half a million of its users, after duly anonymising them. Nevertheless, Narayanan and Shmatikov soon published a research paper [136] showing that, by combining the Netflix Prize data with the public, not anonymised, Internet Movie Database (IMDb), it was easy to reidentify the users behind the ratings. Even if Netflix sincerely thought that its users' privacy was protected, the data release could actually reveal, after reidentification, personal information such as political and sexual orientations, or religious beliefs.

Several other examples of unexpected deanonymisation have been published in the following years - in a telephone database using four geolocation points [57], in a credit card database using four places and dates of transactions [58] - and, nowadays, no serious institution trusts sole anonymisation to protect data.

More recently, in 2019, Garfinkel et al. [75] showed that, using the statistics published by

the U.S. Census, it was possible to reconstruct records of address, age, gender and ethnicity. John Abowd, chief scientist at the U.S. Census Bureau, announced that almost half of the U.S. population records had been reconstructed with this attack [180].

1.4 . Formalisation of privacy

1.4.1 . On the terminology of privacy

In cybersecurity, the term *security* covers three aspects, gathered in the acronym CIA:

- *Confidentiality*: only authorised entities can view sensitive information
- *Integrity*: the data remain intact
- *Availability*: the data are readily available to their users.

Nevertheless, in the common language of cryptographers, the term security is generally used to refer to confidentiality in particular.

In machine learning, the community employs the term *privacy* to refer to the non-disclosure of sensitive information. In the remainder of this manuscript, the term privacy will be used, for the sake of consistency.

1.4.2 . Four shades of privacy

Several notions are associated to privacy, and apply in different contexts:

- *anonymity*: a subject is anonymous from an attacker if the attacker cannot identify the subject within a set of subjects, called the anonymity set
- *unlinkability*: two or more items are said unlinkable if the attacker cannot determine whether they are related or not, given the access to a particular amount of information
- *undetectability*: an item is undetectable if the attacker cannot decide whether it exists or not
- *pseudonymity*: a subject is pseudonymous if it is identified by a pseudonym instead of its real name

As clarified by these concepts, privacy is not necessarily the total absence of information but it has often to do with breaking the links between objects and in particular between a subject's characteristics and the identity of this subject. The first three aspects of privacy can be achieved via DP, a powerful privacy tool that will be introduced in Section 2.1.

1.4.3 . Privacy in cryptography

As early as 1949, Shannon [164] defined the *perfect secrecy* of a cryptosystem as the property that the distribution of probability of the cleartext knowing one of its encryption is equal to the *a priori* distribution of probability of the messages. Goldwasser and Micali [79] took a more computationally oriented point of view and relaxed this definition by restricting it to polynomially bounded adversaries, introducing what they called *semantic security*.

These two notions of security gave birth to the two principal trends in modern cryptography: *information-theoretic security* derived from Shannon's perfect secrecy and *computational security* derived from Goldwasser and Micali's semantic security.

In computational security, the security of a cryptosystem is based on the computational complexity of problems that are conjectured as hard i.e. non-solvable in polynomial time (in $NP \setminus P$) [3].

Goldwasser and Micali's semantic security is equivalent to the satisfaction of what modern cryptographers call *IND-CPA* (for indistinguishability under chosen plaintext attack) for which the adversary is assumed to have access to an oracle (or directly the encryption key in a public-key cryptosystem) that gives it the encryption of any message that it requests. A stronger security class is *IND-CCA* (for indistinguishability under chosen ciphertext attack) whereby the adversary is not only supposed to have access to an encryption oracle but also to an oracle that, given a ciphertext, returns the output of the decryption function, which is not necessarily a valid plaintext if the ciphertext was not the result of an encryption but may release information about the decryption key. IND-CCA is divided into two classes, IND-CCA1 and IND-CCA2 which is stronger. It has been proved that HE cryptosystems cannot be IND-CCA2 (as IND-CCA2 is equivalent to non-malleability) and whether a fully homomorphic encryption cryptosystem can be IND-CCA1 is an open question, at least from a practical point of view. All currently used fully homomorphic encryption cryptosystems are only IND-CPA.

Unfortunately, in computational security, the perfect equality of the *a priori* and *a posteriori* distributions is unreachable and an adversary might derive some information about the message from the ciphertext (for instance by using the encryption oracle with brute force). What needs to be ensured is that, within the computational bounds of the adversary, this amount of information is negligible. Thus, the system is said secure (for IND-CPA or IND-CCA depending on the hypothesis on the adversary's capabilities) if the difference between prior and posterior probabilities, known as the *adversary's advantage*, cannot exceed $2^{-\lambda}$ without performing $O(2^{\lambda})$ computations, where λ is the *security parameter* of the cryptosystem.

[3] If $P \neq NP$, NPC is a strict subset of $NP \setminus P$ that is why the considered problems are not necessarily conjectured to be NP-complete.

2 - Privacy tools

In this section, we present some technical tools that are used to ensure privacy, especially in machine learning, and give some formal background as well as intuitive insights to understand these tools. This presentation does not at all aim at being exhaustive and will focus on the two tools that we employed in our contributions, namely DP and HE.

2.1 . Differential privacy

The word *hazard* originally comes from Arabic az-zahr, the die. According to some authors, az-zahr may come from a word meaning flower, because a flower was drawn on one of the faces of dice. The etymological meaning is then close to the neutral meaning of randomness, that remains in other languages like French or Spanish. Nevertheless, usage has given the English word the derogatory connotation of *risk*, because uncertainty has often worried humans. However, many scientific discoveries have proven randomness useful, and DP, which protects data privacy by delivering noised information to the adversary, is one of them.

DP is defined in a context where an adversary has access to the output of a mechanism and wants to know the input database, hesitating between two databases that differ by only one individual, called the adversary's *victim*. By adding enough random noise to the output of the mechanism, it is possible to "blur" the print of the differing individual that remained in the output so that the adversary will not be able to decide which database was the input with a significant advantage over guessing. A corollary is that the noise obfuscates the information about the victim that the adversary could have retrieved from the output. Figure 2.1 illustrates the training of a model under DP guarantees. Note that the adversary that hesitates between the two databases may know everything about the non-differing individuals.

2.1.1 . Basic definitions

Three years after Nissim and Dinur proved in 2003 that, to be private, a database cannot answer queries exactly but needs to introduce some noise [61], Dwork et al. proposed and formalised a novel paradigm to protect data privacy, known as *differential privacy* (DP) [65]. DP is now a gold standard concept in privacy-preserving data analysis. It provides a guarantee that, under a reasonable privacy cost ϵ, two *adjacent* databases produce statistically indistinguishable results.

The notion of adjacency varies among authors. In our collaborative frameworks, the term database denotes the concatenation of all the agents' datasets and two databases are adjacent if they have the same number of agents and differ on a single agent (being called *teacher* or *client* depending on the situation), all the others remaining unchanged. Yet, the differing agents may have totally different data, making our notion of adjacency quite conservative (this is called *user-level privacy*). Indeed, user-level privacy protects all the data of a single agent, making this kind of privacy stronger than *sample-level privacy* for which two databases are adjacent when they differ on a single sample.

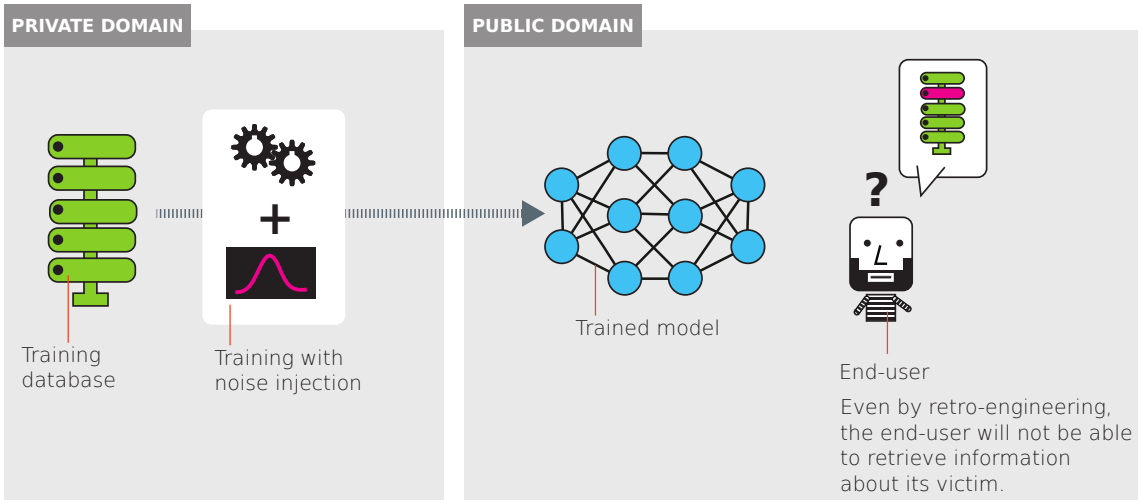

Figure 2.1: Differentially private training. Image reproduced from [168], with the kind authorisation of Florent Robert from Industrie et Technologies.

Let us formally define DP and the privacy cost.

Definition 1 (Pure differential privacy)**.** *Given $\epsilon \in \mathbb{R}_+$, a* probabilistic *mechanism $\mathcal{A}$ with output range $\mathcal{R}$ satisfies (or is)* ϵ-DP *if, for any two adjacent databases d, d' and for any subset $S \subset \mathcal{R}$, one has*

$$\mathbb{P}\left[\mathcal{A}(d) \in S\right] \leq e^{\epsilon}\mathbb{P}\left[\mathcal{A}(d') \in S\right].$$

This notion, rarely applicable in practise, was soon extended to the notion of *approximate differential privacy* [63].

Definition 2 (Approximate differential privacy)**.** *Given $(\epsilon, \delta) \in (\mathbb{R}_+)^2$, a* probabilistic *mechanism $\mathcal{A}$ with output range $\mathcal{R}$ satisfies (or is)* (ϵ, δ)-DP *if, for any two adjacent databases d, d' and for any subset $S \subset \mathcal{R}$, one has*

$$\mathbb{P}\left[\mathcal{A}(d) \in S\right] \leq e^{\epsilon}\mathbb{P}\left[\mathcal{A}(d') \in S\right] + \delta.$$

(ϵ, δ) is called the privacy cost*.*

In the case where $\delta = 0$, we are back to the original notion of Definition 1, that is usually called *pure* DP. Unless the contrary is explicitly stated, we will hereafter refer to approximate DP as, simply, DP.

One easily sees that the lower are ϵ and δ, the closer are the output distributions and the more private is the mechanism. Besides, the mechanism $\mathcal{A}$ cannot be deterministic otherwise the only way to get a finite privacy cost would be to give the same output to two adjacent databases leading, by transitivity, to a constant mechanism that would be totally useless.

Some differentially private (DP) mechanisms are probabilistic by construction (see Chapter 2 from Part II) but most of them derives from deterministic functions on which random noise is applied. One of the most widely used DP mechanism is the *Gaussian mechanism*, which simply

adds a Gaussian noise of mean 0 and standard deviation $\sigma \in \mathbb{R}_+^*$, to each component of the output. *Laplace mechanism*, with Laplace noise, is also frequently used and has the advantage of yielding pure DP guarantees.

A fundamental notion in DP is the *sensitivity*, defined below. When the considered mechanism is obtained by adding random noise on a deterministic function, the sensitivity measures the influence of any single sample (resp. agent) - and thus typically the adversary's victim - on the output of the function. It somehow plays the role of a Lipschitz constant with respect to the adjacency relation.

Definition 3. *Let $\mathcal{A}$ be a randomised mechanism obtained by adding random noise on a deterministic function f. Given a norm $\|\cdot\|$, the $\|\cdot\|$-sensitivity of f is*

$$S = \max_{d,d' \text{ adjacent}} \|f(d) - f(d')\|$$

where the maximum is taken over all pairs of adjacent databases.

The sensitivity is used in works involving DP to scale the added noise (or, more properly, its standard deviation): the more sensitive to the input data the deterministic function is, the more noise we will need to add to protect these data.

Figure 2.2 illustrates the role of sensitivity in a randomised mechanism, namely the Laplace mechanism. Clearly, the greater the sensitivity is, the greater is the shift between the two distributions and the higher is the ratio of probabilities to bound. This ratio is actually constant on $\mathbb{R}$ in the case of the Laplace mechanism. The DP guarantee δ is then null while ϵ is equal to the logarithm of this constant ratio.

Several alternative notions of DP have appeared in the literature [60] to keep track of the privacy cost more tightly or account for the specificity of some problems; among the most famous are concentrated DP [67], Rényi DP [132], computational DP [133]. Authors generally provide connections between these alternative notions and the original definition, such as implications from one notion to the other.

2.1.2 . Differential privacy in examples

To give a more concrete grasp of how DP can be applied in practise, let us present two examples that use quite different techniques.

As a first example, let us explain the randomised response algorithm, described in [66] to introduce DP. A statistical agency conducts a survey in order to estimate the number of illegal drug consumers in a country. Simply asking to each person if he or she consumes illegal drugs seems difficult since many consumers would not want to answer the truth. A solution, actually used in social sciences, is to tell the respondents to answer the truth with a certain probability p, $\frac{1}{2} < p < 1$. In that case, whatever is the respondent's answer, it can never be known with certainty whether this respondent consumes drugs or not. But, if p is large enough, the noised number of consumers obtained from the survey ($(2p-1)c + (1-p)n$ in expectation if n is the total number of respondents and c the number of respondents that consume drugs) is close to the true number of drug consumers.

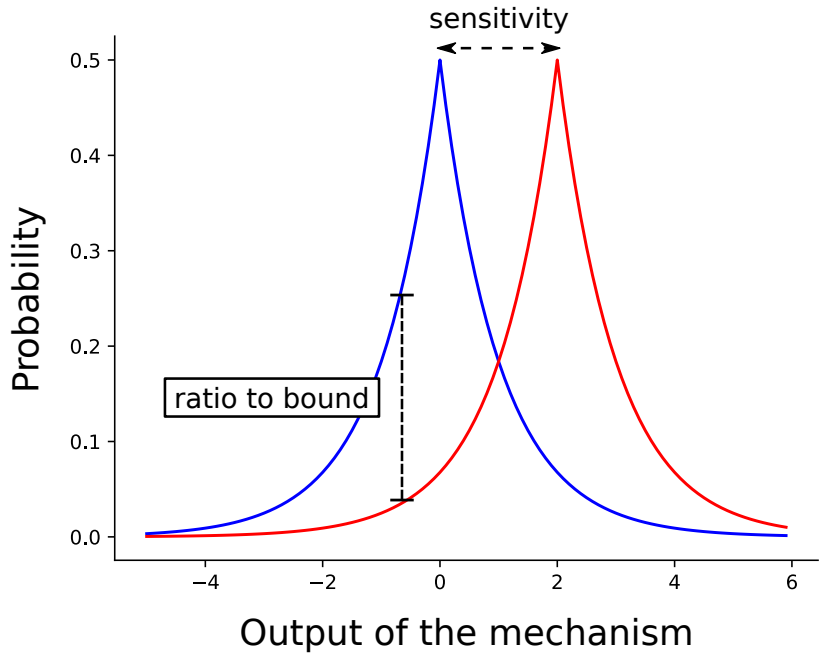

Figure 2.2: Output distributions of Laplace mechanism for two adjacent databases

The second example is closer to most of the works in the literature and the works that will be presented in this manuscript (except Chapter 2 from Part II) and deals with a randomised mechanism constituted by a deterministic function on which a random noise is added. We suppose that one asks for the average age of the inhabitants of a small village of, say, 100 people. It is besides assumed that the asking entity may be an adversary that knows the age of everybody in the village but one person, called its victim. The goal of the adversary's query is to learn its victim's age. Indeed, given the average age, the adversary only needs to multiply it by 100 and subtract the ages of all the villagers except its victim. The result will be its victim's age. If we consider the villagers' ages as sensitive information, we face an issue of privacy violation. Hence, the idea is to noise the average age before delivering it to the querying entity. For instance, we can add a Gaussian noise of standard deviation, say 6 months (i.e. 0.5 year), to the average. Such a noise would not harm the useful information too much, since six months is quite a short time compared to the age of a person. However, the adversary would get, after multiplication by 100 and subtraction of the known ages, its victim's age plus a noise of standard deviation $100 \times 0.5 = 50$ years, which would totally obfuscate the victim's age[1]. The efficiency of DP in this example stems from the gap between the amount of noise added to the sensitive information and the one induced on the useful information. The ratio between the two standard deviations is equal to the number of villagers. Actually, it is a general fact in DP that hiding the information from one single individual among a population becomes easier

[1]To derive actual DP guarantees, one would need to bound the victim's age to get a sensitivity value.

when the population is greater.

Obviously, machine learning is currently one of the main application fields of DP. In most cases, one wants to protect the training data privacy so the mechanism to sanitise is the function that takes the training dataset as input and outputs an inference result. To do so, some noise is added in the training process or at inference time. Since this thesis focuses on this application field, we give further details in a devoted section (Section 2.1.7).

2.1.3 . Almost-omniscient adversary

The classical assumption of DP that the adversary may perfectly know all the individuals of the database except its victim (let us say in this case that the adversary is *almost-omniscient*) might seem a bit exaggerated at first sight. The justification is that it is the most conservative assumption for an adversary that has a victim it wants to learn about, i.e. any adversary that knows less will *a fortiori* be defended against by the DP mechanism. Moreover, making this assumption greatly simplifies the DP analysis in most cases.

Under this assumption of almost-omniscience, one may find superfluous to even give any new information to the adversary since this new DP information is supposed not to reveal anything (or rather not to reveal much) about the victim and hence, could already be deduced by the almost-omniscient adversary. In the example of the average age, the almost-omniscient adversary could obtain a rather accurate average age by computing the average of the ages of all the inhabitants except its victim.

But a DP-protected output can be seen as the common reasonable answer to two types of queries corresponding to two types of queriers, when the data owner does not know which type she is facing:

- a genuine query from someone who does not know much (or anything) about the dataset and wants to have a statistic on it
- a malicious query from an almost-omniscient adversary that wants to infer some information about an individual of the database; in this case the adversary will not get much new useful information from the answer since the noised mechanism is precisely designed to give an output that is practically indistinguishable if the victim was or was not involved

Note that the DP-protected output is also a reasonable answer to any intermediary querier between these two types (typically an adversary that may know something about the database but that is not almost-omniscient).

The data owner wants to give the almost-omniscient adversary the information it already has, and nothing more. Why then not giving it the exact statistic, with all the individuals except its victim as an input? This is because the data owner does not know who is the adversary's victim. The data owner then has to give the adversary an answer that does not leak much information, whoever is its victim.

2.1.4 . Composition across multiple queries

It is clear that the privacy cost that we introduced in Definition 2 is associated to the information of a single drawing of the output distribution and, hence, to a single query. If the

adversary queries the probabilistic mechanism several times, it will get more information about the output distribution and, at the limit of an infinite number of queries, it will be able to reconstruct this distribution exactly and deduce the input database.

That is why, to determine the privacy cost of a protocol, a two-fold approach is traditionally adopted. First of all, one determines the privacy cost per query and, in a second step, one composes the privacy costs of each query to get the overall cost. The classical composition theorem (see e.g. [66]) states that the guarantees ϵ and δ of sequential queries add up. Nevertheless, training a deep neural network, even with a collaborative framework, requires a large amount of calls to the databases, precluding the use of this classical composition. Therefore, to obtain reasonable DP guarantees, one needs to keep track of the privacy cost with a more refined tool. The one that we will use in our contributions is the *moments accountant* [2], that we introduce here.

Definition 4. *The* moments accountant *is defined for any* $l \in \mathbb{N}^*$ *as*

$$\alpha_{\mathcal{A}}(l) := \max_{aux,d,d'} \log \left(\mathbb{E}_{o\sim\mathcal{A}(aux,d)} \left[\left(\frac{\mathbb{P}[\mathcal{A}(aux,d)=o]}{\mathbb{P}[\mathcal{A}(aux,d')=o]} \right)^l \right] \right)$$

where the maximum is taken over any auxiliary input aux *and any pair of adjacent databases* (d, d')*.*

The moments accountant (which is closely related to Rényi DP [132]) allows for a new DP composition. In fact, Theorem 1 shows that the moments accountant of adaptive queries sum up.

Theorem 1 ([2])**.** *Let* $p \in \mathbb{N}^*$*. Let us consider a mechanism* $\mathcal{A}$ *defined on a set* $\mathcal{D}$ *that consists of a sequence of adaptive mechanisms* $\mathcal{A}_1, \ldots, \mathcal{A}_p$ *where, for any* $i \in \{1, \ldots, p\}$*,* $\mathcal{A}_i \colon \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \mapsto \mathcal{R}_i$*. Then, for any* $l \in \mathbb{N}^*$*,*

$$\alpha_{\mathcal{A}}(l) \leq \sum_{i=1}^{p} \alpha_{\mathcal{A}_i}(l).$$

Finally, once δ is chosen, the DP guarantee is derived from the overall moments accountant applying the tail bound property, stated in Theorem 2 from [2]. Note that a mechanism like the Laplace mechanism that gives pure DP guarantees for one query will lose this advantage when the moments accountant method is used for composition.

Theorem 2 ([2])**.** *For any* $\epsilon \in \mathbb{R}_+^*$*, the mechanism* $\mathcal{A}$ *is* (ϵ, δ)*-differentially private for* $\delta = \min_{l\in\mathbb{N}^*} \exp(\alpha_{\mathcal{A}}(l) - l\epsilon)$*.*

In practice, composing the moments accountant and getting back to the standard DP guarantees afterwards gives far better results than the traditional method. For instance, in [2], $\delta = 10^{-5}$ being fixed, the moment accountant technique allows to pass from $\epsilon = 9.34$ to $\epsilon = 1.26$ for the training of a neural network via DP-SGD (Differentially Private Stochastic Gradient Descent) for image classification.

2.1.5 . Post-processing

It is both intuitive and well-known that applying a function, deterministic or probabilistic, does not increase the amount of information: the output contains less (or the same quantity of) information than the input[2]. This translates in the immunity of DP to *post-processing*, as stated below.

Proposition 1 ([66]). *Let $\mathcal{A}$ be a probabilistic mechanism, with output range $\mathcal{R}$, that is (ϵ, δ)-differentially private, with $(\epsilon, \delta) \in (\mathbb{R}_+)^2$. Let ϕ: $\mathcal{R} \to \mathcal{R}'$ be an arbitrary probabilistic mapping. Then $\phi \circ \mathcal{A}$ is (ϵ, δ)-differentially private.*

This property is a key advantage of DP and is widely used in the literature to analyse complex mechanisms.

2.1.6 . Differential privacy and information theory

The concept of DP can be linked to information theory in the sense that it aims at reducing the amount of information about an individual given to the adversary by the output of a mechanism. [177] proposes a tutorial that surveys how current works view DP in the information-theoretic scope, building bridges between DP and tools from information theory like mutual information, min entropy, Kullback-Leibler divergence, rate-distortion function. Among these works, Mir [131] shows that DP arises from information-theoretic optimisations: the mechanism that minimises mutual information between the output and the input, given a distortion rate, is the exponential mechanism, which is DP. Similarly, the mechanism that follows the minimum discrimination information principle, given a distortion rate, is also the exponential mechanism. Using the graph structure properties of the adjacency relation, Alvim et al. [9] prove that (pure) DP implies an upper bound for information leakage (measured by Rényi min entropy) for all the databases and for a single individual.

2.1.7 . Differential privacy in machine learning

DP has become an inescapable tool in machine learning for whoever wants to protect the training data privacy. Indeed, it has been shown that an adversary that has access to the model once it is trained (typically an *end-user*) may retrieve information about the training data, even with a *black box* access [3]. Even if some attacks are possible on generative adversarial networks, variational auto-encoders [47, 90, 94] or other kinds of algorithms such as decision trees, linear and logistic regressions [93, 175, 190], principal component analysis [191] support vector machines and hidden Markov models [10], we will hereafter speak about classical neural networks, which are by far the most targeted machine learning model as shown in the survey of Rigaki et al. [152].

These attacks can be seen as reverse engineering, a sort of incomplete inversion of the function that maps the training database to the trained model, as illustrated by Figure 2.3.

[2] Interestingly, if the applied function is injective, no information is lost.

[3] *Black box* access means that one can only see the inference output of the model when giving it a certain input. On the contrary, a *white box* access assumes that one knows all the parameters of the model (architecture, weights and biases for a neural network).

The incomplete information that can be retrieved can be of different kinds as the following classification of attacks shows:

- *membership inference*: the adversary wants to know if a given sample was part of the training set [36, 95, 119, 158, 167, 175]
- *data reconstruction*: the adversary aims at reconstructing a training sample
  - either reconstructing the totality or unknown features of an actual training sample (*attribute inference*) [37, 139, 140, 195]
  - or generating a typical representative of a given class [73, 96, 193]
- *property inference*: the adversary has access to some information about a data sample and tries to infer a property of this sample that is independent of the learning task [74, 130, 170]

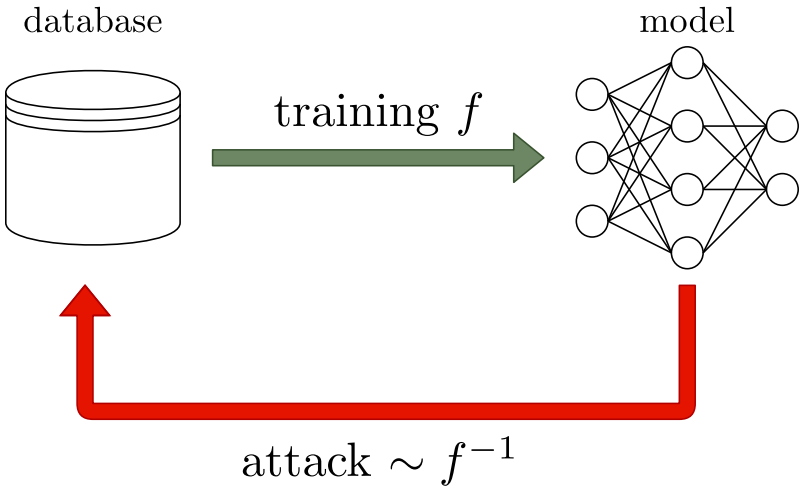

Figure 2.3: Seeing privacy attacks as reverse engineering

Unfortunately, by construction, DP cannot defend against the data reconstruction attacks that generate an average of the samples of the targeted class but only against attacks that focus on one single individual's data, namely membership inference, attribute inference and property inference.

Note that, by Proposition 1, no post-processing involved by these attacks will be able to retrieve more information than stipulated by the DP guarantees.

To defend against these attacks via DP, one has to decide where to add noise in the sequence of computations that constitute the training. The most straightforward solution would be to add the noise on the parameters of the trained model just before releasing it like in the toy example of the average age (Section 2.1.2) However, thanks to the post-processing property (Section 2.1.5), one may turn private any of the successive results that lead to the trained model (or even any of the successive inference steps in the case of a black box access). While most works in deep learning add noise to the gradients, using DP-SGD (differentially private

gradient descent) [2], some authors proposed to noise the output or the objective function via what is known as *output* or *objective perturbation* algorithms [146, 147, 187] or even noise the input features [148]. This choice of the "place" where adding the noise obviously depends on the learning procedure and will highly impact the privacy-utility trade-off. The noise should be added to values whose sensitivity can be tightly bounded so that the level of noise - more formally its standard deviation - can be optimised and set to the smallest value necessary to get the required privacy guarantees.

Indeed, as in any application using DP, there is a trade-off between privacy and utility: the more noise is added, the more protected are the data but the less accurate will be the model. Nevertheless, in machine learning, this trade-off curse is not a fatality. Some authors [15, 103] showed that the DP noise may actually help the model to generalise better, like a sort of regularisation, even if most works see this noise as a threat to model accuracy.

In some cases, when it is an actual actor of the training, an adversary may have access to sensitive information throughout the learning phase, and not only to the trained model as an end-user. For instance, if a server takes part in the training, it will see some data before its own processing, which constitutes a greater information leakage than seeing only the processed data. This threat is common in collaborative learning even if, in many papers whose focus is elsewhere, the server is considered as trustworthy [77, 127, 141, 142, 166]. When it is not, one solution is to employ what is called *local differential privacy* [62, 107, 108]. This consists in the data owner applying enough noise to its data before outsourcing them. Nevertheless, this generally has a strong utility cost [108, 176] especially for deep learning applications. As an illustrative example, let us consider a collaborative setting with $n \in \mathbb{N}^*$ data owners, and let us assume that the data from one owner need a random noise of standard deviation $\sigma \in \mathbb{R}_+^*$ to be protected with the required DP guarantees. Supposing that the noises sampled by the data owners are independent and identically distributed (i.i.d.) and that the server will sum up all the noised values received from the data owners (as in Federated Averaging described in Section 3.1 for example) then, the sum of the values will be noised with a standard deviation of $\sigma\sqrt{n}$. On the contrary, if only the end-users were a threat and the server were trusted, it would have been enough to noise the sum of values, on the server side, with a standard deviation of σ, gaining a factor $\sqrt{n}$ regarding the distortion of the data.

Another solution is to use cryptographic techniques to get rid of the server threat as we will see in Chapter 3. Thanks to such techniques, the server has only access to the aggregation of the values from the clients (or nothing in the case of HE), which is far easier to protect than the individual values. We can link the aggregation trick to the notion of anonymity: since the aggregation operator is permutation-invariant, it provides anonymity to the individual values. It is thus not surprising that the protection entailed by aggregation (gaining a $\sqrt{n}$ factor comparing to local DP) is better than the protection provided by shuffling (gaining a $\sqrt{\log(1/\delta)n}$ factor comparing to local DP [68]).

2.1.8 . Real-life applications

Nowadays, DP has been mainly used in research even if open-source implementations are now available, such as Tensorflow privacy or Opacus. Nevertheless, this powerful tool has started to be used in real-world applications. The US census Bureau used DP on its 2020 Census

(https://www.ncsl.org/health/differential-privacy-for-census-data-explained) which has raised several practical and ethical questions about the impact of that new method on the census, particularly accuracy and fairness issues. Apple collects data about the habits of use of its clients in order to improve the user experience with, e.g., quickType suggestions, emoji suggestions. To do so without jeopardising their clients' privacy, they make use of DP (https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf).

2.2 . Cryptographic primitives

2.2.1 . The homomorphic encryption paradigm

Homomorphic encryption (HE) is a kind of cryptographic method allowing to perform computations over encrypted data without decryption. Let us consider Λ and Ω which respectively are the set of cleartexts (*a.k.a.* the clear domain) and the set of ciphertexts (*a.k.a.* the encrypted domain). As any cryptosystem, a HE system first consists in two algorithms $\mathsf{Enc}_{\mathsf{pk}} : \Lambda \longrightarrow \Omega$ and $\mathsf{Dec}_{\mathsf{sk}} : \Omega \longrightarrow \Lambda$ where pk and sk are data structures which represent the public encryption key and the private decryption key of the cryptosystem. In a symmetric-keyed cryptosystem, the encryption and decryption keys are the same key, which is, obviously, private.

Any (decent) HE scheme possesses the *semantic security* property, equivalent to IND-CPA (cf. Section 1.4.3), meaning that, given $\mathsf{Enc}(m)$ and polynomially many pairs $(m_i, \mathsf{Enc}(m_i))$ it is hard[4] to gain any information on m with a significant advantage over guessing. In particular, the semantic security implies that HE systems are by necessity probabilistic, meaning that some randomness has to be involved in the Enc function and that the ciphertexts set Ω is significantly much larger than the cleartexts set Λ. Indeed, if we consider a deterministic asymmetric system encrypting binary values, an adversary could encrypt both 0 and 1 and compare these encryptions to the encrypted value it observes, leading it to discover the clear value with certainty.

Most importantly, a HE scheme offers two other operators $\oplus$ and $\otimes$ where

- $\mathsf{Enc}(m_1) \oplus \mathsf{Enc}(m_2) = \mathsf{Enc}(m_1 + m_2) \in \Omega$
- $\mathsf{Enc}(m_1) \otimes \mathsf{Enc}(m_2) = \mathsf{Enc}(m_1 m_2) \in \Omega$

Figure 2.4 shows a simple protocol where an HE-blinded server enables two parties to get a function of the union of their data without compromising their privacy.

When these two operators are supported without restriction by a homomorphic scheme, it is said to be a fully homomorphic encryption (FHE) scheme. A FHE scheme with $\Lambda = \mathbb{Z}_2$ is Turing-complete and, as such, is *in principle* sufficient to perform any computation in the encrypted domain with a computational overhead depending on the security target[5]. *In practice*, though, the $\oplus$ and $\otimes$ are much more computationally costly than their clear domain counterparts

[4] "Hard" means that it requires solving a reference (conjectured) computationally hard problem on which the security of the cryptosystem hence depends. From a practical viewpoint, given a security target λ, the concrete parameters of a homomorphic scheme are chosen such that the best known (exponential-time) algorithms for solving the underlying reference problem require an order of magnitude of 2^λ nontrivial operations.

[5] Polynomial in λ.

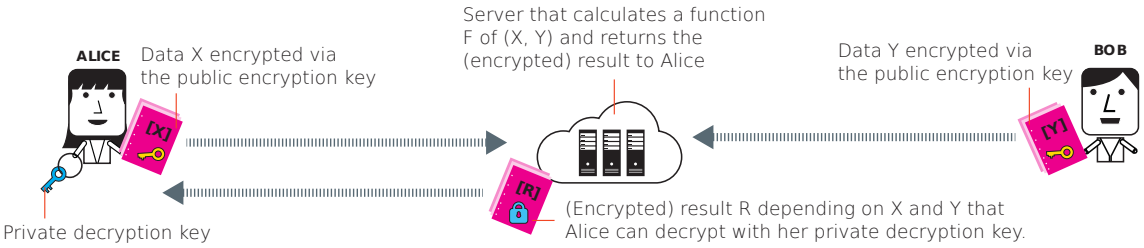

Figure 2.4: Homomorphic encryption. Image reproduced from [168], with the kind authorisation of Florent Robert from Industrie et Technologies.

which has led to the development of several approaches to HE schemes design each with their pros and cons.

2.2.2 . Somewhat homomorphic encryption

Somewhat homomorphic encryption schemes, such as BGV [31] or BFV [69], provide both operators but with several constraints. Indeed, in these cryptosystems the $\otimes$ operator is much more costly than the $\oplus$ operator and the cost of the former strongly depends on the *multiplicative depth* of the calculation, that is the maximum number of multiplications that have to be chained (although this depth can be optimised [11]). Interestingly, most SHE schemes offer a *batching* capability by which multiple cleartexts can be packed in one ciphertext resulting in (quite massively) parallel homomorphic operations i.e.,

$$\mathsf{Enc}(m_1, ..., m_\kappa) \oplus \mathsf{Enc}(m'_1, ..., m'_\kappa) = \mathsf{Enc}(m_1 + m'_1, ..., m_\kappa + m'_\kappa) \tag{2.1}$$

(and similarly for $\otimes$). Typically, several hundreds such slots are available, which often allows to significantly improve the amortised computation time (nevertheless, the latency may remain important).

2.2.3 . Fully homomorphic encryption

Besides the computational cost of the homomorphic operators, the results of homomorphic additions and multiplications are noisy, and this noise increases as more operations are applied. As a consequence, the number of operations that can be computed in the encrypted domain before the result is totally useless is limited. To solve this fundamental issue, Gentry proposed the *bootstrapping* technique in 2009 [76]. By designing a homomorphic cryptosystem that is able to evaluate its own decryption circuit, the noised ciphertexts can be refreshed and the level of noise maintained bounded, for any number of operations. Therefore, such a cryptosystem can theoretically perform any function in the encrypted domain, giving birth to FHE.

FHE schemes offer both the $\oplus$ and $\otimes$ operators without restrictions on multiplicative depth. At the time of writing, only the FHE-over-the-torus approach, instantiated in the TFHE cryptosystem [50], offers practical performances. In this cryptosystem, $\oplus$ and $\otimes$ have the same constant cost. On the downside, TFHE offers no batching capabilities.

2.2.4 . Homomorphic encryption in practice

The bottom line of xHE calculations *practice* is to use the most appropriate cryptosystem for the problem at hand. Low (multiplicative-)depth calculations (say 5 and below) are generally

performed more efficiently by means of SHEs while higher (multiplicative-)depth ones (say above 5) requires resorting to the full-blown FHE machinery of TFHE. Furthermore, to get the best of all worlds, the TFHE scheme is often hybridised with SHE by means of operators allowing to homomorphically switch among several ciphertext formats [27, 120] to perform each part of calculation with the most appropriate scheme (see *e.g.* [197]).

Additionally, it should be emphasised that the natural threat model against HE is limited to *honest-but-curious* adversaries. Such an adversary, i.e. the server which performs the encrypted-domain calculations, properly executes the task it is in charge of but attempts by all possible passive means to retrieve information about the private data values. Therefore, in a machine learning context, it may make use of any data it legitimately has access to, directly or indirectly by performing some (polynomial-time) computations on them, in order to retrieve some information about the private data, yet without harming the learning (or inference) process.

2.2.5 . Homomorphic encryption for machine learning

Most of the works applying HE to machine learning models focus on the inference stage [28, 41, 42, 78, 91, 99, 121, 160] and not on the training stage. The first papers on privacy-preserving machine learning training focused on a centralised setting where all data are outsourced and where the models are only linear [19, 38, 109]. When it comes to non-linear models, the few approaches that ran a complete centralised training of neural networks on encrypted data have impractical performances or huge cryptographic parameters [120]. One of the biggest challenges, both at inference and training stages, is to homomorphically compute the activation functions of the model. This can be achieved by low-degree polynomial approximation [48, 100, 144] or discretisation (splitting the activation functions in several affine parts) [172] but this is time-consuming and the accuracy is degraded.

A way to address this computational issue is to leverage the knowledge of a model that was already trained on a public dataset and fine-tune it with the new dataset, using *transfer learning* techniques [120, 123].

Other frameworks like the one presented in Chapter 3 may also be employed in order to concentrate the homomorphic computations on a core of very few aggregating operations, thus reducing to the minimum the computations evaluated in the encrypted domain.

2.2.6 . Other cryptographic primitives

Several other cryptographic primitives have been used in machine learning literature to protect data privacy. In particular, Secure Multi-Party Computation is a general approach that enables several parties to collaboratively perform a given computation without revealing to the other parties any more information than the result of this computation.

For instance, additive secret sharing allows to distribute to n parties shares of a secret randomly noised and whose sum is equal to the secret. Hence, knowing the totality of the shares is required to reveal the secret, and no strict subset of the shares gives any information about the secret (see e.g. [56]).

Shamir's secret sharing [163] has a more general utility since it allows any subset of $t \leq n$ parties, t being a fixed parameter, to reveal the secret with their shares, while preventing any subset of size strictly smaller than t from knowing the secret.

While they are often less computationally intensive than HE, a major drawback of both additive and Shamir's secret sharings is their lack of malleability. Additive secret sharing only supports addition, and Shamir's secret sharing does not allow to perform operations directly on encrypted values. Moreover, contrarily to HE, the model itself is not hidden from the server, which is required in many application scenarios.

Although these approaches are very close in intent to FHE-based ones they achieve different trade-offs. While FHE is computationally intensive and non-interactive, Secure Multi-Party Computation puts more stress on protocol interactions. Indeed, these techniques require a lot of information exchanges (garbled circuit generation and evaluation, oblivious input key retrieval, secret key sharing), which consume time and bandwidth. Moreover, they generally fail when some parties do not play their role - or fixing the fault tolerance issue implies additional rounds of communication [24,25]. On the contrary, the FHE approach is more versatile, requires no interaction among the teachers and is robust to temporary client unavailability, except in the multi-key setting.

Closer to HE is functional encryption [26], which gathers cryptosystems with a public key and a master secret key. For a function f, a specific secret key can be generated using the master secret key. Applied on a ciphertext c encoding a plaintext x, this specific secret key enables to output $f(x)$ without revealing any additional information on x. A third party that generates the master secret key is usually required but some authors proposed decentralised versions of functional encryption [52,53]. A major drawback of functional encryption is that, in practice, the only functions it supports are inner products [52] and quadratic functions [154], and composition of functions is not possible. For more complex functions, it is much more computationally intensive than FHE.

At the intersection of secret sharing and functional encryption is *function secret sharing* [30]. This technique extends secret sharing by allowing several parties to compute encrypted shares of a function f. When a third-party receives these shares, it is able to output the result of f applied on all the parties' values, without learning anything else than this result. Furthermore, any strict subset of the function shares is insufficient to retrieve f. Interestingly, Ryffel et al. [153] extended 2-party function secret sharing to train a model via federated learning.

While these cryptographic primitives are of great interest, HE has specific features that make it more appropriate to certain problems and, as such, it deserves attention in its own right. That is why this PhD focuses on HE and the contributions presented in Part II will use this primitive.

2.2.7 . Comparison of cryptographic security and differential privacy

The notions of privacy enforced by cryptography on one hand and DP on the other hand seem to be of different natures. In fact, while cryptography ensures an (almost) perfect secrecy, thus making the ciphertext unusable without decrypting it, DP provides useful statistics about a group that do not reveal too much information about any member of the group. However, the difference between these two privacy paradigms can actually be seen as a quantitative difference, rather than a qualitative one. Letting decryption aside, cryptography and DP occupy two different positions of the privacy-utility continuum represented in Figure 2.5.

Thus, it is the very fact that DP allows some information leaking that makes it more

Figure 2.5: The continuum of privacy-utility trade-off

accurate. It appears that this information leaking, quantified by the privacy cost in DP, can actually be formally related to the security parameter in cryptography as shown in the following.

On the one hand, let us consider a binary problem where, given a ciphertext c, the adversary needs to decide if c is the encryption of 0 or 1. The adversary's advantage is then defined as $|p_{success} - p_{max}|$ where $p_{success}$ is the probability that the adversary guesses the true plaintext that c encrypts and p_{max} is the probability of the most probable plaintext *a priori* i.e. the probability of guessing with success without access to c (since the adversary will bet on the most probable answer). Then, if $\lambda \in \mathbb{R}_+^*$ is the security parameter of the cryptosystem,

$$p_{success} - p_{max} \leq 2^{-\lambda}. \tag{2.2}$$

On the other hand, let us consider an ϵ-DP mechanism $\mathcal{A}$, $\epsilon \in \mathbb{R}_+^*$, and an adversary that sees an output o of $\mathcal{A}$ and hesitates between (only) two possible inputs: the adversary needs to decide whether $o = \mathcal{A}(D)$ or $o = \mathcal{A}(D')$, where D and D' are two adjacent databases. Since $\mathcal{A}$ is ϵ-DP, aux being the auxiliary information of the adversary, $e^{-\epsilon} \leq \frac{\mathbb{P}[\mathcal{A}(D)=o|aux]}{\mathbb{P}[\mathcal{A}(D')=o|aux]} \leq e^{\epsilon}$.

We have

$$\begin{aligned}\frac{\mathbb{P}[\mathcal{A}(D) = o|aux]}{\mathbb{P}[\mathcal{A}(D') = o|aux]} &= \frac{\mathbb{P}[\mathcal{A}(x) = o|x = D, aux]}{\mathbb{P}[\mathcal{A}(x) = o|x = D', aux]} \\ &= \frac{\mathbb{P}[x = D|\,\mathcal{A}(x) = o, aux]\mathbb{P}[x = D'|aux]}{\mathbb{P}[x = D'|\,\mathcal{A}(x) = o, aux]\mathbb{P}[x = D|aux]}\end{aligned}$$

(via Bayes formula)

Without loss of generality, we assume that $\mathbb{P}[x = D|\,\mathcal{A}(x) = o, aux] \geq \mathbb{P}[x = D'|\,\mathcal{A}(x) = o, aux]$. Consequently, the adversary that wants to maximise its chance of success will bet on D as an input and we get that the probability of success is $p_{success} = \mathbb{P}[x = D|\,\mathcal{A}(x) = o, aux]$. Moreover, thanks to the auxiliary information, the adversary knows that the input is either D or D' then $\mathbb{P}[x = D'|\,\mathcal{A}(x) = o, aux] = 1 - \mathbb{P}[x = D|\,\mathcal{A}(x) = o, aux] = 1 - p_{success}$. We then get

$$\frac{\mathbb{P}[\mathcal{A}(D) = o|aux]}{\mathbb{P}[\mathcal{A}(D') = o|aux]} = \frac{p_{success}}{1 - p_{success}} \times \frac{\mathbb{P}[x = D'|aux]}{\mathbb{P}[x = D|aux]}$$

and thus

$$\frac{p_{success}}{1 - p_{success}} \times \frac{\mathbb{P}[x = D'|aux]}{\mathbb{P}[x = D|aux]} \leq e^{\epsilon}. \tag{2.3}$$

We can now compare inequalities 2.2 and 2.3 and remark that, in 2.2, the constraint on $p_{success}$ is additive whereas it is multiplicative in 2.3.

To go further, let us assume that, in both contexts, the two answers are *a priori* equally probable according to the adversary. Then, Inequation 2.2 becomes

$$p_{success} - \frac{1}{2} \leq 2^{-\lambda} \iff p_{success} \leq 2^{-\lambda} + \frac{1}{2}$$

and Inequation 2.3 becomes

$$\begin{aligned} \frac{p_{success}}{1 - p_{success}} \leq e^{\epsilon} &\iff \frac{1}{1 - p_{success}} - 1 \leq e^{\epsilon} \\ &\iff p_{success} \leq 1 - \frac{1}{e^{\epsilon} + 1}. \end{aligned}$$

Hence, $1 - \frac{1}{e^{\epsilon}+1}$ plays in DP the role of $2^{-\lambda} + \frac{1}{2}$ in cryptography. Solving the equation, we get

$$\epsilon = \log\left(\frac{1}{\frac{1}{2} - 2^{-\lambda}} - 1\right)$$

and, for λ approaching infinity:

$$\epsilon \underset{\lambda \to +\infty}{\sim} 2^{2-\lambda} = 4 \times 2^{-\lambda}.$$

Hence, we can make the approximation that the DP guarantee ϵ of a cryptosystem decreases exponentially with its security parameter λ.

Given that traditional values for ϵ in DP are usually greater than one (or, at best, slightly lower than one), and that the recommended security parameter by the CNIL (Commission Nationale de l'Informatique et des Libertés), the French national commission for informatics and liberty, is 128, 192 or even 256 bits, it is easy to understand the huge gap between privacy-utility trade-offs chosen in cryptography and DP.

Another difference is that, as far as DP is concerned, the adversary is assumed to make a limited number of queries whereas, in cryptography, the adversary is only bounded in terms of computational complexity (even if some settings impose a constant number of operations).

Note that, except for its local version, DP also leverages the dilution of individual information within a group to improve the privacy guarantees. The guarantee ϵ decreases in $\sqrt{n}$ in a group of size n. Hence, a mechanism satisfying local DP with $\epsilon = 1$ and an accuracy a would need, to keep the same accuracy a with the security guarantees of a cryptosystem with a 128-bit security parameter, to dilute the individual information in a group of size around $\frac{1}{(2^{-126})^2} = 2^{252} \approx 10^{76}$ i.e. only 10000 times less than the estimated number of atoms in the universe !

3 - Collaborative learning

When several data owners want to train a global model that use the knowledge of all their data, a naive approach is to make all the owners send their data to the server and let the server perform the training. This obviously creates severe issues of:

- communication: the transfer of the raw data from the owners to the server is highly demanding in terms of bandwidth and in general the raw data are much heavier than the models. In particular, this precludes the use of HE[1].
- computational resources: having the server perform all the training requires much computing power (the use of is precluded in this case, as explained in Section 2.2.5)
- privacy: if they deem their data as sensitive, the data owners may not be willing to send them entirely or partially to the server

In the following, we will present some frameworks that enable several data owners to jointly train a global model, often with the help of an aggregation server but in a more subtle way that addresses all or part of the aforementioned issues (even, as we shall see in part II, with HE in the picture). These techniques are gathered under the wide term *collaborative learning*. One of the most popular and widely studied in the current literature is *federated learning*.

3.1 . Background on federated learning

With the emergence of smartphones, researchers wondered how to use the huge amounts of new data such as text messages to train machine learning models. The naive option of sending all the data to a central server and letting it perform the whole training is not practical as explained above. In 2016, researchers from Google introduced a new training paradigm addressing this problem, *federated learning* [126].

Federated learning (FL) is a decentralised framework, illustrated in Figure 3.1, that enables multiple agents, called *clients*, to collaboratively train a shared global model under the orchestration of a central server while keeping the training data localised on the client devices. After a common (server-side) arbitrary initialisation of the global model, the FL process consists of successive rounds of communication between the server and the clients. At every iteration, a fraction of the clients receive the current model parameters and update them by minimising a local loss function. Then, they sent the updates back to the server that aggregates them to refresh the global model parameters. This method allows the training data to stay on the client

[1]Transciphering [35] can solve this communication problem. The data owners encrypt their data with a symmetric-key cryptosystem which results in negligible size overhead compared to plaintexts. The encrypted data are sent to the server that encrypts them with a HE cryptosystem and perform the symmetric-key decryption homomorphically. Yet, since this homomorphic decryption is very costly, transciphering basically converts the communication overhead in a computational overhead on the server side.

devices thus helping to protect data privacy and reducing the communication costs comparing to the centralised solution since only the updates are sent to the server. Moreover, the computations are parallelised among all the clients, resulting in a reduction of the wallclock duration of the whole training.

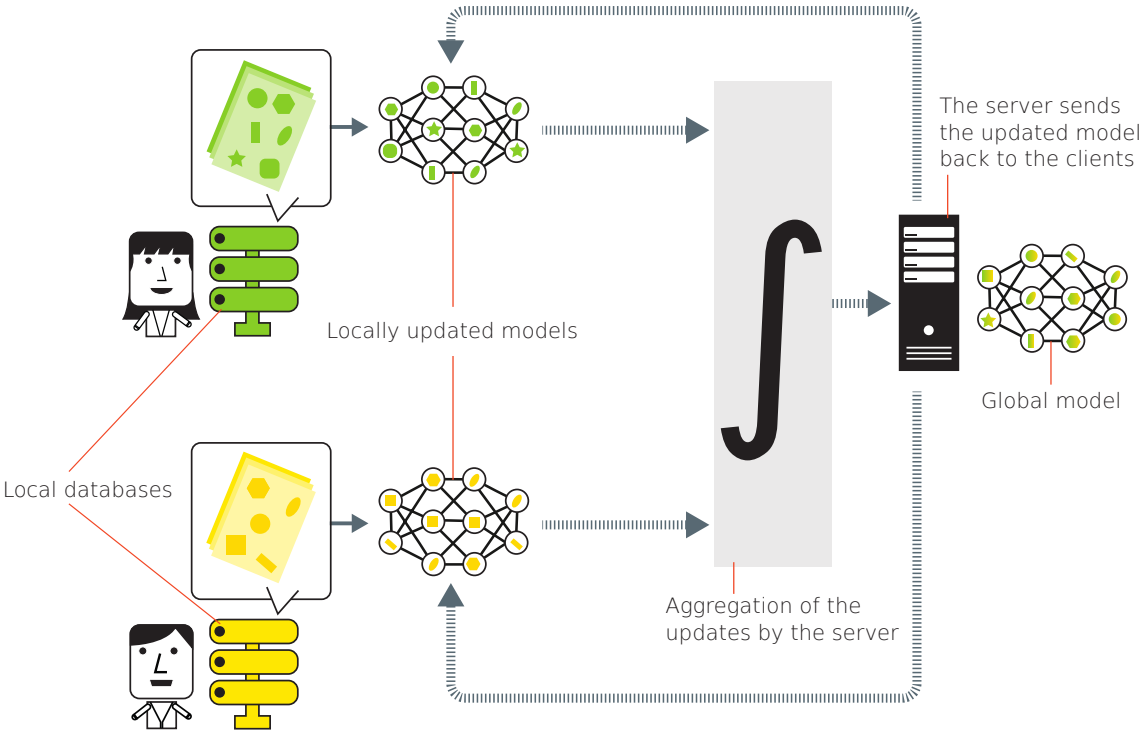

Figure 3.1: Federated learning scheme. Image inspired from [168], with the kind authorisation of Florent Robert from Industrie et Technologies.

A typical application is the training of next-word prediction algorithms on smartphones [89] for which the training data - texts from smartphones users- are both very numerous and private. This is an example of *cross-device* federated learning i.e. a framework that involves a large number of clients (thousands or even millions) that have relatively few and unbalanced data each, with a restricted computational power. On the contrary, *cross-silo* federated learning deals with much fewer clients (less than 1000 in general) that have a large amount of data and computational resources - the clients in cross-silo federated learning are typically companies or institutions.

The most common approach to optimisation for FL is the Federated Averaging algorithm [128] (see Algorithm 1), also known as FedAvg. At the beginning of each round, the server selects a subset of clients to take part in training for this round, we call these particular clients the *participants*. The server sends the current global model to the participants and each of them trains the model locally with several epochs of mini-batch stochastic gradient descent (SGD) using its own data. The participants then communicate only the updated parameters

or the updates[2] themselves (depending on the setting) back to the server. Finally, the server computes the weighted average of these updates before accumulating them into the global model, thereby concluding the round. The weight associated to a participant in the average is generally the fraction of training samples owned by the participant.

FedAvg reaches similar accuracy to the one of centralised setting in the case of i.i.d. (independent identically distributed) data. On the contrary, if the clients' data do not all have the same distribution, the convergence is degraded [105, 182]. Nevertheless, the advantages in terms of communication, computational resources and privacy are generally considered as a fair compensation to this loss as the current popularity of federated learning shows.

Other aggregation rules than the average may be used, for example aggregation rules that are robust to Byzantine attacks, such as Krum, the median, the trimmed mean [70, 118].

Algorithm 1: Federated Averaging (FedAvg)

Server executes:
Input : M: total number of clients
K: number of participants per round
n_k: number of data points of participant k
w_t: model parameters at round t
Output: model parameters w_{t+1} at final round
initialise w_0;
for *each round* t **do**
  $K_t \leftarrow$ random set of $K \leq M$ clients;
  for *each client* $k \in K_t$ *in parallel* **do**
    $u_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$;
  $w_{t+1} \leftarrow w_t + \sum_{k=1}^{K} \frac{n_k}{n} u_{t+1}^k$ where $n = \sum_{k=1}^{K} n_k$

ClientUpdate(k, w)**:**
Input : k: id number of the participant
D_k: training set of participant k
B: local mini-batch size
E: number of local epochs
η: learning rate
w: global model parameters
L: local loss function
Output: updates $w_k - w$ after last epoch
initialise $w_k = w$;
$\mathcal{B} \leftarrow$ split the n_k samples of D_k into batches of size B;
for *each epoch from* 1 *to* E **do**
  for *each batch* $b \in \mathcal{B}$ **do**
    $w_k \leftarrow w_k - \eta \nabla L(w_k; b)$;

[2] Difference between the updated parameters and the old ones.

3.2 . Privacy-preserving collaborative learning

3.2.1 . Distributed differential privacy

One of the crucial issues of privacy-preserving collaborative learning is the threat of a distrusted server that has access to many data throughout the training phase. As mentioned in Section 2.1.7, local DP is an option. Nevertheless, we saw that it often requires to apply too much noise, at the expense of utility. To recover the centralised setting's privacy-utility trade-off, an idea is to make use of cryptographic techniques to hide the data from the server while it is computing them [4, 135].

In this context, two options are possible. The model can remain hidden from the server after training, in that case there is no information leakage on the training data from the server's point of view. Otherwise, if the server has access to the trained model, the noise can obviously not be generated by the server since there is no DP guarantee towards an entity that knows the noise that has been sampled to protect the data. A common solution is called *distributed differential privacy* [63]: the data owners sample and add noise themselves to their data so that after the aggregation by the server, the resulting noise is sufficient to ensure the required privacy guarantees. The easiest example is probably the distributed Gaussian mechanism that uses the stability by addition of the normal law. If there are $n \in \mathbb{N}^*$ and the server is to add the values it receives (like in FedAvg), every data owner applies a noise of standard deviation $\frac{\sigma}{\sqrt{n}}$ to its data so that the sum of the values is noised with a standard deviation of $\sigma \in \mathbb{R}_+^*$. Some works combined distributed DP with secure aggregation techniques like additive secret sharing [3, 80, 165] or *secure shuffling* [22, 49, 68] to get rid of the assumption of a trusted server.

In the case of additive secret sharing, the values to be sent by the clients and summed up by the server are noised with both the DP noise and the secret sharing mask. This implies that, beforehand, the clients used a protocol to collaboratively determine individual masks that sum up to zero. At aggregation, the masks cancel out and the summed DP noises result in the desired amount of noise that ensure the required DP guarantees.

Secure shuffling resorts to the ESA method - Encode, Shuffle, Analyse - where a data are sent encrypted to a shuffler that receives them into batches, eliminates the metadata and shuffles the batches. The shuffler then sends the data to an analyser that decrypts them and performs the required computations. Shuffling allows anomymity and unlinkability of the data (see Section 1.4.2 for the definitions) thus allowing for a better privacy-utility trade-off that lies between the local DP trade-off and the central DP one (without shuffler), while avoiding the assumption of a trusted server.

Nevertheless, these methods are demanding in terms of communication and require an additional trusted entity to generate and send the random masks to the clients or shuffling the clients' contributions. Also note that secure shuffling yields a worse privacy-utility trade-off than secure aggregation.

3.2.2 . Fault tolerance

When implementing distributed DP, the risk of a data owner not showing up and thus preventing correct noise generation arises. To address this problem of *fault tolerance*, some authors

make the server generate the noise that some users did not generate [13] while others assume that the data owners themselves adapt the noise they generate to the possible failures [44]. These works imply that the server, respectively the data owners, know who or at least how many data owners did not generate the noise. Another solution is to make the data owners send a second noise, which will be used in case of default of another data owner [185] but this increases the communication cost. One may also accept this risk of default and analyse how the DP guarantees evolve when a certain fraction of the data owners do not add noise [156], as we did in our contributions from Chapters 1 and 2 from Part II. With such an analysis, one can also determine how much extra noise has to be added per data owner to guarantee a fixed privacy cost even if a certain ratio of data owners do not add noise.

3.2.3 . Federated learning gets along well with homomorphic encryption

In federated learning, and more specifically when FedAvg is employed, the server computations are limited to a simple (possibly weighted) sum, which is quite easy to implement for various cryptographic techniques (one-time pads, additive secret sharing, HE). That is why several authors have used such techniques with federated learning in a context of sensitive user data [24, 25, 149, 153, 192]. The clients encrypt their local updates (or directly the gradients) of the model's parameters obtained by gradient descent and send them to the server which perform the aggregation of the encrypted value (homomorphically or on the masked values). The result of the aggregation is sent back to the clients and decrypted by them in the case of HE (already clear in the case of additive secret sharing).

Closer to our contributions, some works have combined cryptographic primitives and DP [45, 155, 156]. Yet, in this context of iterative protocols and large number of parties, classical multi-party computation techniques suffer from their important communication requirements [25, 134]. HE gets the most out of the game because it needs much less communication rounds than most of the other cryptographic primitives. Its relative computational heaviness is not such a burden here because FedAvg only needs additive cryptosystems, which are much less computationally intensive than fully homomorphic schemes, especially in the case of a large multiplicative depth. Along with distributed DP, a few papers propose the use of HE to protect the clients' data from the server [88, 174] but do not take into account the quantisation of the DP noise due to encryption. In Chapter 3 from Part II, we present a contribution about privacy-preserving federated learning with use of distributed DP and that focuses on this noise quantisation.

3.2.4 . Alternatives to federated learning

Decentralised federated learning: To solve the issue of the untrustworthy server, an interesting alternative to classical federated learning is *decentralised federated learning* that simply gets rid of the server. The information spreads across the clients following a graph whose nodes are the clients and whose edges represent communication channels between clients: a client can only exchange data with its neighbours in the graph. Two approaches are possible to update the model parameters: random-walk based stochastic gradient descent [54] and gossip-based stochastic gradient descent [55, 156, 157]. The graph structure is crucial to this framework and, with a well chosen graph, it is possible to match the centralised federated

learning's privacy-utility trade-off in both approaches (up to a logarithmic factor in the number of clients in [54, 55]). Nevertheless, [156] and [157] resort to secure aggregation that requires a central coordinator and thus is not compatible with full decentralisation. As far as [54] and [55] are concerned, the authors use a relaxation of standard DP which makes their privacy guarantees less conservative and vulnerable to colluding clients and eavesdroppers. Moreover, they provide a mean privacy loss and no explicit guarantees on the maximum privacy loss, as standard DP provides. Finally, their privacy-utility trade-off holds for a learning process that takes a very high number of iterations - scaling with the square of the number of clients in the random-walk approach - and makes the convergence very slow.

Private Aggregation of Teacher Ensembles (PATE): In [141] and [142], the authors proposed a quite different method to privately aggregate the knowledge of several data owners. Each data owner trains a local model with its own data, called a teacher model. Assuming the existence of a public unlabelled database, all the teacher models vote to label the samples of the public database and the most frequent class is chosen, via plurality rule. The public database labelled in this way is used to train a global model, called the student. This method has the advantage of being agnostic to the type and architecture of both the teacher and student models. Besides, they obtain good data-dependent DP guarantees thanks to a detailed analysis of the noisy argmax that is used to implement private plurality. Yet, their approach requires the server to be trusted and, in particular, not curious. We build upon this framework in our contributions of Chapters 1 and 2 from Part II, with an honest-but-curious server.

Part II

Contributions

1 - SPEED: Secure, PrivatE, and Efficient Deep learning

Abstract We introduce a deep learning framework able to deal with strong privacy constraints. Based on collaborative learning, differential privacy and homomorphic encryption, the proposed approach advances state-of-the-art of private deep learning against a wider range of threats, in particular the honest-but-curious server assumption. We address threats from both the aggregation server, the global model and potentially colluding data holders. Building upon distributed differential privacy and a homomorphic argmax operator, our method is specifically designed to maintain low communication loads and efficiency. The proposed method is supported by carefully crafted theoretical results. We provide differential privacy guarantees from the point of view of any entity having access to the final model, including colluding data holders, as a function of the ratio of data holders who kept their noise secret. This makes our method practical to real-life scenarios where data holders do not trust any third party to process their datasets nor the other data holders. Crucially the computational burden of the approach is maintained reasonable, and, to the best of our knowledge, our framework is the first one to be efficient enough to investigate deep learning applications while addressing such a large scope of threats. To assess the practical usability of our framework, experiments have been carried out on image datasets in a classification context. We present numerical results that show that the learning procedure is both accurate and private.

N.B.: This chapter is the reproduction of the article *SPEED: secure, PrivatE, and efficient deep learning*, joint work with Rafaël Pinot, Martin Zuber, Cédric Gouy-Pailler and Renaud Sirdey, published in Machine Learning journal, via ECML-PKDD journal track 2020 [83].

1.1 . Introduction

Application scenarios. We consider n hospitals, each of which owns a (personal) labelled database composed of medical records from its patients and a model (e.g. neural network) trained on this database to predict if a new patient is victim of a given disease, say cancer. The hospitals' goal is to collaborate in order to improve the early detection of cancer. Building a model from a larger dataset than the personal databases would lead to improved detection capabilities. Nevertheless, these medical databases are highly-sensitive and the information they contain about the patients cannot be disclosed [143]. In such a setting, the hospitals wish to collaboratively train a global model while preserving confidentiality of their records. To do so, the idea is to rely on an aggregating institution (e.g. the World Health Organisation). This would amount to creating a three-party architecture: hospitals, aggregating institution, global model. Note that in our example, and in many real-world settings, all the training data providers may be recipients of the global model, or the global model may even be totally public. Hence, the global model may be exposed to attacks like membership inference attacks [167] that

could indicate with high accuracy the probability that one patient was present in a database. Also, given a set of instances, the risk of a model inversion attack [186] which tries to infer sensitive attributes on the instances from a supposedly non-sensitive (often white-box) access to the model, is to be seriously taken into account as it would allow to infer for example that some of the hospital databases contain more ill patients than others. Besides, the aggregating institution might be the target of cyberattacks aimed at stealing data from it. For all these reasons, the three-party architecture we consider has to be resistant to threats coming from *both the aggregation server and the global model recipients*.

Another motivating example, from the field of cybersecurity, is when several actors each hold a database of cybersecurity incident signatures that have occurred on their customer networks. The actors would rely on a third-party server to train the global model. In this scenario, it is a great security issue if the global model suffers from an attack (e.g. if the model features can be inferred [173, 181, 188] with limited access to the model). In this case, this would clearly leak some information on the detection capabilities of the actors, giving a clear advantage to cyberattackers on the networks they supervise.

Deployment scenario and threat model. To perform the aggregation in a private way, we work in the tripartite setting summarised in Figure 1.1 and formally detailed in Section 1.4. The *student* (who holds the global model, a.k.a. the *student model*) is the owner of the homomorphic encryption scheme under which encrypted-domain computations will be performed by the *aggregation server*. This means that the student generates and knows both the encryption and decryption keys pk and sk. Then, when being submitted an unlabelled input, the data holders (a.k.a. the *teachers*) noise the predictions from their personal models, encrypt them under pk and send these encryptions to the server. The server has the responsibility to homomorphically perform the aggregation in order to produce an encryption of the output (e.g. a label) which will be sent back to the student and used by the latter for learning, after due decryption. *Homomorphic encryption* thus provides a countermeasure to confidentiality threats on the teachers' predictions from the aggregation server, while the noise introduced by the actor addresses, via *differential privacy*, the issue of attacks against the student model. In this setting, we assume that the student model is public or at least available to all the actors of the protocol, namely the teachers, the aggregation server and, of course, the student. Our mechanism is differentially private in this context, and our guarantees still hold against a malicious teacher, who has the information of the noise she generated, or even against colluding teachers (see Section 1.5). On the contrary, we do not address threats whereby the student and the aggregation server collude in the sense that the student does not share sk with the server (in which case they would both get access to the teachers' predictions). We do not consider either threats where the aggregation server behaves maliciously, e.g. to prevent the student model from effectively learning from the teachers, leading to more or less stealthy forms of denial-of-service, or to perform a chosen ciphertext attack via selected queries to the student model. This is the typical scenario in which homomorphic encryption intervenes and our setting thus covers the threat model whereby the aggregation server is assumed to operate properly but may perform computations on observed data to retrieve information. This threat model is

commonly known as the *honest-but-curious* model [24, 82, 98].

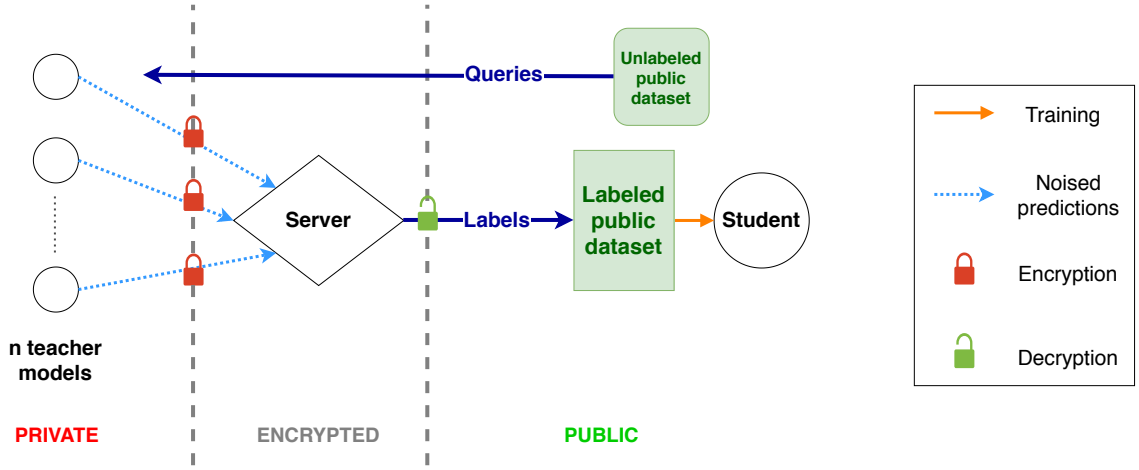

Figure 1.1: SPEED - Teacher models send to the aggregation server their encrypted noisy answers to the student's queries. The server homomorphically performs the aggregation in the encrypted domain and sends the result to the student model which decrypts it and uses it for training

Our contribution. In this paper, we present a complete collaborative learning protocol which is secure along the whole workflow regarding a large scope of threats. We ensure protection of the data against any malicious actor of the protocol during the learning phase and prevent indirect information leakage from the final model using *both* homomorphic encryption and differential privacy. While our framework is agnostic to the kind of models used by both the teachers and the student, to the best of our knowledge this is the first work with this level of protection to be efficient enough to apply to deep learning, therefore allowing very good accuracy on difficult tasks such as image classification, as shown by the experiments we ran. Our framework is also bandwidth-efficient and does not require more interactions than required by the baseline protocol.

Outline of the paper. Section 1.2 relates our work to the literature. In Section 1.3, we give some technical background on differential privacy and homomorphic encryption. We describe our SPEED framework in Section 1.4 and analyse its differential privacy guarantees in Section 1.5. Section 1.6 presents our experimental results - SPEED achieves state-or-the-art accuracy and privacy with a mild computational overhead w.r.t previous works. Section 1.7 concludes the paper and states some open questions for further works.

1.2 . Related work

Differential privacy (DP). Recent works considered to use differential privacy in collaborative settings close to the one we consider [17, 21, 45, 77, 141, 142]. Among them, the most efficient technique in terms of accuracy and privacy guarantees is Private Aggregation

of Teacher Ensembles (PATE) first presented in [141] and refined in [142]. PATE uses semi-supervised learning to transfer to the student model the knowledge of the ensemble of teachers by using a differentially private aggregation method. This approach considers a setting very close to ours with the notable difference that the aggregation server is trusted. Hence, applying PATE in our scenario makes the teacher models vulnerable. To tackle this issue, our work builds upon PATE idea with two key differences: we let the responsibility of generating the noise to the teachers and we add a layer of homomorphic encryption in order for the overall learning to be kept private. Another difference can also be noted. To derive privacy guarantees, PATE assumes that two databases d and d' are adjacent if only one sample of the personal database d_i of one teacher i changes, with the hypothesis that the personal databases d_i are disjoint. We do not need this hypothesis and we only consider the teacher models, not the personal databases they use to train them. This leads us to a more powerful definition of adjacency: two databases d and d' are adjacent if they differ by one teacher.

Homomorphic Encryption (HE). HE allows to perform computations over encrypted data. In particular, this can be used so that the model can perform both training and prediction without handling cleartext data. In terms of learning, the naive approach would be to have the training sets homomorphically encrypted, sent to a server for training to be done in the encrypted domain and the resulting (encrypted) model sent back to the participants for decryption. However, putting aside many subtleties, even by deploying all the arsenal available in the HE practitioner toolbox (batching, transciphering, etc.) this would be impractical as "classical" learning is both computation and know-how intensive and HE operations are intrinsically costly. As a consequence, there are only very few works that capitalise on HE for private training [82, 92, 120] and inference [78, 104] of machine learning tasks. Moreover, since some attacks can be performed in a black-box setting, the system is still vulnerable to attacks from the end user who has access to the decryption key. In our framework, we do not use HE directly to build the model, we use it as a mean for the aggregation to be kept private. That way, we are protected against potential threats from the aggregation server, which does not have the decryption key, and we keep a manageable computational overhead.

Federated learning. Federated learning approaches gather several users who own data and make them collaborate in an iterative workflow in order to train a global model. The most famous federated learning algorithm is federated averaging [126] which is a parallelised stochastic gradient descent. In a context of sensitive user data, several works proposed privacy-preserving federated learning or closely related distributed learning that make use of differential privacy [77, 166], cryptographic primitives [24, 25, 153] or both [45, 155, 156]. These methods require online communication between the parties whereas our solution takes advantage of homomorphic encryption and the existence of personal trained models to avoid online communication and drastically limit the interactions, that are both bandwidth-consuming and vulnerable to attacks.

Private aggregation. Several approaches have been considered to limit the need for a trusted server when applying differential privacy, for example by considering local differential privacy [62, 107, 108]. In practice it often results in applying too much noise, and maintaining utility can be difficult [108, 176] especially for deep learning applications. In order to recover more accuracy while keeping privacy, some works combined decentralised noise distribution (*a.k.a.* distributed differential privacy [165]) and encryption schemes [3, 80, 150, 165] in the context of aggregation of distributed time-series. Our work contributes to this line of research. However, our framework is the first one to be efficient enough to investigate deep learning applications while combining distributed DP and HE. Another advantage of our solution concerns fault tolerance regarding the added noise. Some works addressed the problem of fault tolerance by making the server generate the noise that some users did not generate [13] while other works assume that the users themselves adapt the noise they generate to the possible failures [44]. In our setting, because of the encryption and the absence of communication between the teachers, we cannot suppose that any honest entity knows if some failures occurred. Moreover, the addition of noise to compensate a failure does not solve the problem of colluding teachers who may still send noise but do not keep it secret. In our protocol, the task of an honest actor (teacher or server) does not depend on the number of failures and we provide privacy guarantees as a function of the number of failures (see Section 1.5) - it then suffices to assume an upper bound on this number to ensure a privacy guarantee.

Secure Multi-Party Computation (SMPC). Secure Multi-Party Computation is a general approach that enables several parties to collaboratively perform a given computation without revealing to the other parties any more information than the result of this computation. In particular, secure aggregation regroups approaches which use SMPC techniques as one-time pads masking [24, 25] or secret-sharing [56] to perform aggregation over sensitive data. Although these approaches are very close in intent to FHE-based ones, as the present one, they achieve different trade-offs. In a nutshell, when FHE is computation-intensive and non-interactive, SMPC puts more stress on protocol interactions. SMPC requires a lot of communication (garbled circuit generation and evaluation, oblivious input key retrieval, secret key sharing), both time-consuming and vulnerable to attacks, and needs in general that *all* teachers play their role in the protocol for it to terminate - or fixing the fault tolerance issue implies additional rounds of communication [24, 25]. On the contrary, the FHE approach is more versatile, requires no interaction among the teachers and is robust to temporary teacher unavailability. Still, at the time of writing, it is the authors' opinion that both approaches are worth investigating in their own right (and this paper obviously belongs to the FHE thread of research).

1.3 . Preliminaries

1.3.1 . Differential privacy

Let us also present a famous and widely used differentially private mechanism, known as the *report noisy max* mechanism.

Definition 5. *Let $K \in \mathbb{N}^*$, and let $\mathcal{X}$ be a set that can be partitioned into K subsets $\mathcal{X}_1, \ldots,$*

$\mathcal{X}_K$. *The mechanism that, given a database* d *of elements of* $\mathcal{X}$*, reports* $\operatorname{argmax}_{k\in[K]}[n_k + Y_k]$*, where* $[K] := \{1, \dots, K\}$*,* $n_k := |d \cap \mathcal{X}_k|$ *and* Y_k *is a Laplace noise with mean* 0 *and scale* $\frac{1}{\gamma}$ *(with probability density* $x \mapsto \frac{\gamma}{2}e^{-\gamma|x|}$*),* $\gamma \in \mathbb{R}_+^*$*, is called* report noisy max.

Theorem 3 ([66]). *Let* $\mathcal{A}$ *be the report noisy max as above. Then* $\mathcal{A}$ *is* $(2\gamma, 0)$*-differentially private.*

We now define the notion of *infinite divisibility* that we will use to implement distributed differential privacy.

Definition 6. *A random variable* Y *is said to be* infinitely divisible *if, for any* $m \in \mathbb{N}^*$*, we can find a family* $(X_{m,i})_{i\in[m]}$ *of independent and identically distributed (i.i.d.) random variables such that* Y *has the same distribution as* $\sum_{i=1}^m X_{m,i}$*.*

The following proposition from [112] claims that the Laplace distribution is infinitely divisible [1], enabling to distribute its generation among an arbitrary number of agents.

Proposition 2 ([112]). *Let* $m \in \mathbb{N}^*$ *and* $\gamma \in \mathbb{R}_+^*$*. Let* $G_p^{(i)}$*, for* $(i, p) \in [m] \times [2]$*, be i.i.d. random variables following the Gamma distribution of shape* $\frac{1}{m}$ *and scale* $\frac{1}{\gamma}$*. Then* $\sum_{i=1}^m \left(G_1^{(i)} - G_2^{(i)}\right)$ *follows the Laplace distribution of mean* 0 *and scale* $\frac{1}{\gamma}$*. The Laplace distribution is said to be* infinitely divisible*.*

1.4 . SPEED: Secure, Private, and Efficient Deep Learning

1.4.1 . A distributed learning architecture

Let us consider a set of n owners (a.k.a. *teachers*) each holding a personal sensitive model f_i. We assume that we also have an unlabelled public database D. The goal is to label D using the knowledge of the private (teacher) models to train a collaborative model (a.k.a. *student model*) mapping an input space $\mathcal{X}$ to an output space $[K] = \{1, \dots, K\}$. To do so while keeping the process private, we follow the setting illustrated by Figure 1.1 relying on a (distrusted) aggregation server:

1. For every sample x of the public database D, the student sends x to the aggregator requesting it to output a label for x. The aggregator forwards this request to the n teachers.

2. Each teacher i labels x using its own private model f_i. Then each teacher adds noise to the label (see Section 1.4.2) and encrypts the noisy label before sending it to the aggregation server.

[1] Another well-known example of infinitely divisible probability distribution is the Gaussian distribution which can be seen as the sum of Gaussian distributions of well chosen scale parameter. In a possible further work, we could indeed replace the (distributed) Laplace noise by a (distributed) Gaussian noise.

3. The aggregator performs a homomorphic aggregation of the noisy labels and returns the result to the student model, namely the most common answered label (see Section 1.4.3).

4. The student, who owns the decryption key, decrypts the aggregated label and is then able to use the labelled sample to train its model.

Our framework addresses two kinds of threats using two complementary tools. On the one hand, differential privacy protects the sensitive data from attacks against the student model. Indeed, some model inversion attacks [186] might disclose the training data of the student model, and especially the labels of database D. But differential privacy ensures that the noise applied to the teachers' answers prevents the aggregated labels from leaking information about the sensitive models f_i [2]. On the other hand, the homomorphic encryption of the teachers' answers prevents the aggregator to learn anything about the sensitive data while enabling it to blindly compute the aggregation.

1.4.2 . Noise generation and threat models

When requested to label a sample x, each owner i uses its model f_i to infer the label of x. In order for the aggregator to compute the most common label in the secret domain, the owner must send a one-hot encoding of the label. That is, rather than sending $f_i(x)$, the i-th teacher sends a K-dimensional vector, say $z^{(i)}$, whose $f_i(x)$-th coordinate is an encryption of 1 while all the other coordinates are encryptions of 0. To guarantee differential privacy (see Section 1.5 for the formal analysis), the owner adds to this one-hot encoding a noise drawn from $G_1^{(i)} - G_2^{(i)}$ where the $G_1^{(i)}$ and $G_2^{(i)}$ are $2n$ i.i.d. K-dimensional random variables following the Gamma distribution of shape $\frac{1}{n}$ and scale $\frac{1}{\gamma}$, where $\gamma \in \mathbb{R}_+^*$. Then, i sends the (encrypted) noisy one-hot encoded vector whose k-th coordinate corresponds to $z_k^{(i)} + G_{k,1}^{(i)} - G_{k,2}^{(i)}$.

Assuming that the aggregator has access to the student model, distributing the responsibility of adding the noise among all the teachers instead of delegating this task to the aggregator (see paragraph on centralised noise below) is necessary to protect the data against an honest-but-curious aggregator. Indeed, such an aggregator could use the information of the noise it generated to break the differential privacy guarantees and, potentially, recover the sensitive data by model inversion on the student model. Note that such an attack does not break the honest-but-curious assumption since the aggregator still performs its task correctly.

Beyond the honest-but-curious model In a model that would go beyond the honest-but-curious aggregator hypothesis, the capability for an aggregator to add its own noise is even more harmful for the privacy (and of course, the accuracy) than not using noise at all, if the aggregator has access to the student model after training. Indeed it gives the aggregator much more freedom to attack. As an example, think about a malicious aggregator that wants to know a characteristic χ on a particular teacher, called its victim. Given a query, for all $k \in [K]$, we write $n_k := |\{i : f_i(x) = k\}|$ and call it the number of *votes* for class k. Let us suppose that, for a given query, changing the value of the victim's characteristic χ from χ_0 to χ_1 also changes

[2] Thanks to the DP guarantees, the labels of D could actually be published as well.

the victim's vote from a class k_0 to a class k_1. Hence, by denoting $n_{k_0} = \nu_0$ and $n_{k_1} = \nu_1$ if $\chi = \chi_0$, we get $n_{k_0} = \nu_0 - 1$ and $n_{k_1} = \nu_1 + 1$ if $\chi = \chi_1$. Then, if the aggregator knows all the n_k for $k \in [K] \setminus \{k_0, k_1\}$ and knows ν_0 and ν_1 (which are the classical hypotheses in differential privacy), it can add just as much noise as needed for the class k_0 to be the argmax if and only if $\chi = \chi_0$ [3]. The result from the homomorphic argmax would then leak the information about the value of the victim's characteristic χ.

Centralised noise generation In a context in which the student model is kept private and, especially, not available to the aggregator, we can consider a centralised way of generating the noise. If we do not trust the teachers to generate the noise, we can charge the aggregator to do it, since it will not be able to use the knowledge of the noise to attack the sensitive data via the student model. The aggregator only needs to generate a Laplace noise (in the clear domain), and homomorphically add it to the unnoisy encryption of n_k it receives from the teachers. The infinite divisibility of the Laplace distribution (Proposition 2) shows that the resulting noise is the same as in the case presented above in which each teacher generates an individual noise drawn from the difference of two Gamma distributions. The privacy cost of one request is simply the privacy cost of the *report noisy max*, namely 2γ (Theorem 3).

In a nutshell, we can consider the following different threat models regarding the server:

- honest (H) : the aggregation server performs its tasks properly and do not try to retrieve information from the data it has access to
- honest-but-curious (HBC) : the aggregation server performs its tasks properly but it may compute the available data to get sensitive information
- beyond honest-but-curious (BHBC) : the aggregation server performs the aggregation correctly but cannot be trusted to properly generate the noise necessary to the DP guarantees. Note that this threat model is only slightly beyond the honest-but-curious model since the honesty of the server is only relaxed regarding the noise generation, but not the aggregation.

Table 1.1 summarises against which kind of server our protocol is protected, depending on the access the server has to the student model and on the way the noise is generated. As already emphasised, in the following we focus on the case whereby the student model is public and the noise is distributively generated by the teachers because it is the most general model among the realistic threat models and thus gives the better trade-off between flexibility and security.

1.4.3 . Technical details on the homomorphic aggregation

Summing the noisy counts The aggregation server receives the n encrypted noisy labels and sums them up in the secret domain. Due to the infinite divisibility of the Laplace

[3] For example, add $\nu_0 - \frac{1}{2} - n_k$ to all the classes except k_0 and k_1, $\nu_0 - 1 - \nu_1$ to the class k_1 and nothing to the class k_0.

Table 1.1: Robustness of our framework depending on the availability of the student model and the noise generation

	Private model	Public model
Centralised noise	**HBC**	**H**
Distributed noise	**BHBC**	**BHBC**

distribution, the server obtains a K-dimensional vector whose k-th ($k \in [K]$) coordinate is an encryption of:

$$\sum_{i=1}^{n} \left(z_k^{(i)} + G_{k,1}^{(i)} - G_{k,2}^{(i)} \right) = n_k + Y_k$$

where $n_k := |\{i : f_i(x) = k\}|$ and Y_k is a Laplace noise with mean 0 and scale $\frac{1}{\gamma}$.

So far, we have only needed homomorphic addition which is a good start. Then an argmax operator must be performed after the summation. However, *efficiently* handling the highly nonlinear argmax function by means of FHE is much more challenging.

$$\begin{aligned} M(i,2^l) &= M(i,2^{l-1}) \otimes \theta(N(i,2^{l-1}), N(i+2^{l-1},2^{l-1})) \\ &\quad \oplus M(i+2^{l-1},2^{l-1}) \otimes (1 \ominus \theta(N(i,2^{l-1}), N(i+2^{l-1},2^{l-1}))) \\ N(i,2^l) &= N(i,2^{l-1}) \otimes \theta(N(i,2^{l-1}), N(i+2^{l-1},2^{l-1})) \\ &\quad \oplus N(i+2^{l-1},2^{l-1}) \otimes (1 \ominus \theta(N(i,2^{l-1}), N(i+2^{l-1},2^{l-1}))) \end{aligned}$$

Computing the argmax. Most prior work on secure argmax computations use some kind of interaction between a party that holds a sensitive vector of values and a party that wants to obtain the argmax over those values. The non-linearity of the argmax operator presents unique challenges that have mostly been handled by allowing the two interested parties to exchange information. This means increased communication costs and, in some cases, information leakage. This is with the exception of [197]. They provide a fully non-interactive homomorphic argmax computation scheme based on the TFHE encryption. We implemented and parameterised their scheme to fit the specific training problems presented in Section 1.6. We present here the main idea behind this novel FHE argmax scheme. For more details, see the original paper. The TFHE encryption scheme provides a *bootstrap* operation that can be applied on any scalar ciphertext. Its purpose is threefold: switch the encryption key; reduce the noise; apply a non-linear operation on the underlying plaintext value. This underlying operation can be seen as a function

$$g_{t,a,b}(x) = \begin{cases} a & \text{if} \quad x > t \\ b & \text{if} \quad x < t. \end{cases}$$

One notable application is that of a "sign" bootstrap: we can extract the sign of the input with the underlying function $g_{0,1,0}(x)$. The argmax computation in the ciphertext space is made as follows. For every k, k', $k \neq k'$, we compare the values $n_k + Y_k$ and $n_{k'} + Y_{k'}$ with a subtraction $(n_k + Y_k - n_{k'} - Y_{k'})$ and application of a sign bootstrap operation. This yields $\theta_{k,k'}$, a variable

with value 1 if $n_k + Y_k > n_{k'} + Y_{k'}$ and 0 otherwise. Therefore the complexity will be quadratic in the number of classes. For a given k we can then obtain a Boolean truth value (0 or 1) for whether $n_k + Y_k$ is the maximum value. To this end, we compute

$$\Theta_k = \sum_{i \neq k} \theta_{k,i}.$$

n_k is the max if and only if, for all i one has $\theta_{k,i} = 1$ *i.e.* $\Theta_k = K - 1$. We can therefore apply another bootstrap operation with $g_{K-\frac{3}{2},1,0}$. If $\Theta_k = K - 1$, the bootstrap will return an encryption of 1, and return an encryption of 0 otherwise. Once decrypted, the position of the only non-zero value is the argmax. Because the underlying function $g_{t,a,b}$ is applied homomorphically, its output is inherently probabilistic. In the FHE scheme used, an error is inserted in all the ciphertexts at encryption time to ensure an appropriate level of security. This means that if two values are too close, then the sign bootstrap operation might return the wrong result over their difference. The exact impact of this approximation on the accuracy is evaluated in Section 1.6.

Remark. Another solution would be to send the noisy histogram $n_k + Y_k$ of the counts for each class k to the student and let her process the argmax in the clear domain. This could indeed be performed with a plain-old additively-homomorphic cryptosystem such as Paillier or (additive-flavoured) ElGamal, avoiding the machinery of the homomorphic argmax. Nevertheless, this approach was put aside because sending the whole histogram instead of the argmax would provide much worse DP guarantees.

1.5 . Differential privacy analysis

In this section, we will give privacy guarantees considering that two databases d and d' are adjacent if they differ by one teacher i.e. there exists $i_0 \in [n]$ such that $f_{i_0} \neq f'_{i_0}$ and, for all $i \in [n] \setminus \{i_0\}$, $f_i = f'_i$. This definition of adjacency is quite conservative and is strictly larger than the definition of adjacency from [141] (indeed, in the assumption whereby the personal teacher databases d_i are disjoint, changing one sample from a personal database changes at most one teacher).

Robustness against colluding teachers. We have decided not to trust the aggregation server to generate the noise necessary to the privacy guarantees. Hence, we may also assume that a subset of teachers might be malicious and collude by communicating their generated noise. This gives the same DP guarantees from the point of view of a colluding teacher as if they would have not generated any noise. To this extent, our protocol, which addresses this issue, is fault tolerant. The following theorem quantifies the privacy cost of such failures.

In the following, we call $\mathcal{A}$ the aggregation mechanism that outputs the argmax of the noisy counts. $\mathcal{A}(d, Q)$ is the output of $\mathcal{A}$ for the database d and the query Q. Let $\gamma \in \mathbb{R}_+^*$ be the inverse scale parameter of the distributed noise. Considering the DP guarantees from the point of view of an entity $\mathcal{E}$, let $\tau \in (0, 1)$ be the ratio of the teachers whose noise is ignored by $\mathcal{E}$.

Theorem 4. *Let us define* $I\colon v \in \mathbb{R}_+^* \mapsto \int_0^{+\infty} (t+v)^{\tau-1} t^{\tau-1} e^{-2t} dt$ *and* $g\colon t \in \mathbb{R} \mapsto \frac{\int_{\gamma t}^{+\infty} e^{-v} I(v) dv}{\int_{\gamma(t+2)}^{+\infty} e^{-v} I(v) dv}$.

Then, from $\mathcal{E}$'s point of view, $\mathcal{A}$ is $(\epsilon, 0)$-differentially private, with

$$\epsilon = \log\left(1 + 2\frac{\int_0^{\gamma} e^{-v} I(v) dv}{\int_{2\gamma}^{+\infty} e^{-v} I(v) dv}\right).$$

Moreover, if $\tau > \frac{1}{2}$, g is differentiable in 0 and $\mathcal{A}$ is $(\epsilon', 0)$-differentially private, with

$$\epsilon' = \min\left[\epsilon, \log\left(g(0) - g'(0)\right)\right]$$

where $g'(0) = \gamma \frac{\frac{\Gamma(\tau)^2}{2} e^{-2\gamma} I(2\gamma) - I(0) \int_{2\gamma}^{+\infty} e^{-v} I(v) dv}{\left(\int_{2\gamma}^{+\infty} e^{-v} I(v) dv\right)^2}$.

Sketch of proof: Adapting the proof of the privacy cost of the report noisy max from [66], we first show that, if we can find a function M of γ and τ such that, for any $t \in \mathbb{R}$, $g(t) \leq M$, then $\mathcal{A}$ is $(\log(M), 0)$-differentially private. This motivates us to find an upper bound of g.

To do so, we prove that g has a maximum on $\mathbb{R}$ and that this maximum is reached on the interval $[-1; 0]$. On one hand, we show that, for all $t \in [-1; 0]$, $g(t) \leq 1 + 2\frac{\int_0^{\gamma} e^{-v} I(v) dv}{\int_{2\gamma}^{+\infty} e^{-v} I(v) dv}$. On the other hand, we prove that, if besides $\tau > \frac{1}{2}$, then g is concave on $[\operatorname{argmax}(g); 0]$ and thus, for all $t \in [-1; 0]$, $g(t) \leq g(0) - g'(0)$ (note that g is not differentiable in 0 if $\tau \leq \frac{1}{2}$). □

Let us denote S the subset of teachers who are honest (i.e. do not collude). Assuming that the colluding teachers do add noise but communicate it among them, this theorem allows us to control the privacy cost by the ratio τ of the teachers who kept their noise secret, from the point of view of both:

- a colluding teacher, taking $\tau = \frac{|S|}{n}$
- an honest teacher, taking $\tau = \frac{n-1}{n}$
- any entity who has access to the student model but is not a teacher, taking $\tau = 1$

Note that we can also use Theorem 4 in the hypothesis whereby the colluding teachers publish their noise (to the whole world), adapting τ in consequence [4]. For $\tau = 1$, the privacy guarantee is given by $\lim_{\tau \to 1} \epsilon'$ which, as shown by Proposition 3, is the classical bound of the *report noisy max* with a centralised Laplace noise.

Proposition 3. *For all $\gamma \in \mathbb{R}_+^*$, $\lim_{\tau \to 1} \left[\log(g(0) - g'(0))\right] = 2\gamma$.*

Furthermore, Proposition 4 shows that, naturally, the privacy cost tends to be null when the noise becomes infinitely large (γ approaches 0).

[4] e.g. the privacy guarantee for an honest teacher would be computed with $\tau = \frac{|S|-1}{n}$.

Proposition 4. *For all* $\tau \in (0,1)$, $\lim_{\gamma \to 0} \left[\log \left(1 + 2 \frac{\int_0^{\frac{\gamma}{2}} e^{-v} I(v) dv}{\int_\gamma^{+\infty} e^{-v} I(v) dv} \right) \right] = 0.$

Let us also give an upper bound of the probability that the noisy argmax is different from the true argmax.

Proposition 5. *Let* k^* *be the class corresponding to the true argmax.*

If $\tau \in (\frac{1}{2}; 1)$,

$$\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \leq \sum_{k \neq k^*} e^{-\gamma \Delta_k} \left[\frac{1}{2} + \frac{(\gamma \Delta_k)^{2\tau - 1}}{\tau 2^{4\tau - 2} \Gamma(\tau)^2} \right]$$

where $\Delta_k := n_{k^*} - n_k$ *for any* $k \in [K]$ *and* $\Gamma : \beta \in \mathbb{R}_+^* \mapsto \int_0^{+\infty} t^{\beta - 1} e^{-t} dt$ *is the gamma function.*

If $\tau \in (0; \frac{1}{2}]$,

$$\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \leq \sum_{k \neq k^*} e^{-\gamma \Delta_k} \left[\frac{1}{2} + \frac{(\gamma \Delta_k)^{\frac{\tau}{2}}}{\tau 2^{\frac{5}{2}\tau - 1} \Gamma(\tau)^2} \times \left(\frac{3}{2} \tau \right)^{\frac{3}{2}\tau} \left(\frac{2}{\tau} - 3 \right)^{1 - \frac{3}{2}\tau} \right].$$

Sketch of proof: The event $(\mathcal{A}(d;Q) \neq k^*)$ is the union of the events $(n_k + Y_k \geq n_{k^*} + Y_{k^*})$, for $k \in [K] \setminus \{k^*\}$, and thus $\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \leq \sum_{k \neq k^*} \mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*})$. We remark that, for any $k \in [K] \setminus \{k^*\}$,

$$\begin{aligned} &\mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*}) = \mathbb{P}(Y_{k^*} \leq Y_k - \Delta_k) \\ &\quad = \int_{-\infty}^{0} f(t) F(t - \Delta_k) dt + \int_0^{\Delta_k} f(t) F(t - \Delta_k) dt + \int_{\Delta_k}^{+\infty} f(t) F(t - \Delta_k) dt \end{aligned}$$

where $f : u \in \mathbb{R}^* \mapsto \frac{\gamma}{\Gamma(\tau)^2} e^{-\gamma |u|} I(\gamma |u|)$ and $F : t \in \mathbb{R} \mapsto \int_{-\infty}^t f(u) du$.

We show that $\int_{\Delta_k}^{+\infty} f(t) F(t - \Delta_k) dt \leq \frac{3}{8} e^{-\gamma \Delta_k}$ and $\int_{-\infty}^0 f(t) F(t - \Delta_k) dt \leq \frac{1}{8} e^{-\gamma \Delta_k}$. Moreover, using Hölder's inequality, we show that, for all $q \in (\frac{1}{1-\tau}; +\infty)$, calling $p := \frac{1}{1 - \frac{1}{q}}$, $\int_0^{\Delta_k} f(t) F(t - \Delta_k) dt \leq \frac{e^{-\gamma \Delta_k}}{\tau 2^{4\tau - 2 + \frac{1}{q}} \Gamma(\tau)^2} \times \frac{(\gamma \Delta_k)^{2\tau - 1 + \frac{1}{q}}}{p^{\frac{1}{p}} [q(1-\tau) - 1]^{\frac{1}{q}}}$. For $\tau > \frac{1}{2}$, we take the particular (and classic) case of the limit of the previous bound when q tends to $+\infty$. For $\tau \leq \frac{1}{2}$, we take $q = \frac{1}{1 - \frac{3}{2}\tau}$. □

Theorem 4 and Proposition 5 serve as building blocks to which we apply the following theorem from [141].

Theorem 5 ([141]). *Let* $\epsilon, l \in \mathbb{R}_+^*$. *Let* $\mathcal{A}$ *be a* $(\epsilon, 0)$*-differentially private mechanism and* $q \geq \mathbb{P}[\mathcal{A}(d) \neq k^*]$ *for some outcome* k^*. *If* $q < \frac{e^\epsilon - 1}{e^{2\epsilon} - 1}$, *then for any additional information* aux *and any pair* (d, d') *of adjacent databases,* $\mathcal{A}$ *satisfies*

$$\alpha_{\mathcal{A}}(l; aux, d, d') \leq \min \left[\epsilon l, \frac{\epsilon^2 l(l+1)}{2}, \log \left((1-q) \left(\frac{1-q}{1 - e^\epsilon q} \right)^l + q e^{\epsilon l} \right) \right].$$

As in [141], Theorem 5 coupled with some properties of the moments accountant (composability and tail bound) allows one to devise the overall privacy budget (ϵ, δ) for the learning procedure (see Section 1.6 for numerical results). We refer the interested reader to Section A of the appendix for more details and for the extended proofs of our claims.

Influence of the cryptographic layer. One must be aware that the cryptographic layer perturbs the noisy votes because the computation of the homomorphic argmax has a small probability of error. Although this topic deserves further investigations, we make the assumption that these perturbations are negligible and that they do not change the privacy guarantees as they basically constitute an additional noise on the votes. We further discuss this point in Appendix A.3.

1.6 . Experimental results

The experiments presented below enable us to validate the accuracy of our framework on well-known image classification tasks and illustrate the practicality of our method in terms of performance, since the computational overhead due to the homomorphic layer remains reasonable. The source codes necessary to run the following experiments are available on https://github.com/Arnaud-GS/SPEED.

HE time overhead. We implemented the homomorphic argmax computation presented in Section 1.4.3. Without parallelising, a single argmax query for 10 classes and 250 teachers requires just under 4 seconds to compute on an Intel Core i7-6600U CPU. Importantly, this does not depend on the input data. The costliest operation is the computation of θ. Any other part of the scheme is negligible in comparison. Therefore, once the parameters are set, the time performance depends solely on the number of classes (the number of bootstrap comparisons is quadratic in the number of classes). As such, 100 queries require 6.5 minutes and 1000 queries 65 minutes. Of course, the queries can be performed in parallel to decrease the latency allowing for much more challenging applications.

Homomorphic argmax accuracy. As we mention in Section 1.4.3, the homomorphic computation of the argmax is inherently probabilistic. This is due both to the noise added to any ciphertext at encryption time, and to limitations of the bootstrapping operation in terms of accuracy. On MNIST dataset [113], we evaluate the method with $\tau = 1/0.9/0.7$ and compare the cleartext argmax to our homomorphic argmax. Our implementation of the HE argmax has an average accuracy of 99.4%, meaning that it retrieves the cleartext argmax 99.4% of the time.
To obtain a more general and conservative measure of the inherent accuracy of the HE argmax (which can be applied on any dataset), we make the teachers give uniformly random answers to the queries. In this setting, most counts n_k are likely to be close to one another, which makes even a classical argmax useless. This kind of scenario can be seen as worst-case, since the teacher voting is adversarial to argmax computation. Even in this scenario, and with the

same parameters as for MNIST, our implementation of the HE argmax algorithm still produces an average accuracy of 90%. Hence, an accuracy of 90% can be considered a lower bound for any adaptation of this argmax technique to other datasets. Yet in practice a tweaking of the parameters can yield a better accuracy even for this worst-case scenario, at the cost of time efficiency.

Learning setup. To evaluate the performances of our framework, we test our method on MNIST [113] and SVHN [137] datasets. To represent the data holders, we divide the training set in 250 equally distributed and disjoint subsets, keeping the test set for learning and evaluation of the student model. Then we apply the following procedures. We refer the interested reader to Section C of the appendix for more details on the hyper-parameters and learning procedure.

- *Teacher models.* For MNIST, given a dataset, a data holder builds a local model by stacking two convolutional layers with max pooling and a fully connected layer with ReLu activations. Two additional layers have been added for SVHN.
- *Student model.* Following the idea from [141], we train the student in a semi-supervised fashion. Unlabelled inputs are used to estimate a good prior distribution using a GAN-based technique first introduced in [159]. Then we use a limited amount of queries (100 for MNIST, 500 for SVHN) to obtain labelled examples which we use to fine tune the model.

For MNIST experiments, as the student model can substantially vary based on the selected subset of labelled examples, the out-of-sample accuracy has been evaluated 15 times, with 100 labelled examples sampled from a set of 9000 ones. For each experiment, the remaining 1000 examples have been used to evaluate the student model accuracy. For SVHN, the computations being much more heavy, the out-of-sample accuracy has been evaluated 3 times, with 500 examples sampled from a set of 10000 ones. We used 16032 examples to test the student model accuracy.

Performances on MNIST. Table 1.2 displays our experimental results for SPEED with MNIST and compares them to a non-private baseline (without DP or HE) and to the framework that we call Trusted which assumes that the server is trusted and thus only involves DP and not HE. Trusted can be considered as PATE framework from [141] with some subtle differences: the noise is generated in a distributed way in Trusted and the notion of adjacency is larger. Even if the inverse noise scale γ we use is greater than the one in [141] (0.1 instead of 0.05), which should lead to a worse DP guarantee, an argmax-specific analysis of the privacy cost per query allowed us to provide a better DP guarantee ($\epsilon = 1.41$ instead of $\epsilon = 2.04$ with $\delta = 10^{-5}$ and 100 queries). To be more conservative in terms of accuracy, the experiments were run considering that the colluding teachers did not generate any noise, which does not change anything in terms of DP. That is why, in spite of the variability of the accuracy, we observe a trade-off between accuracy and DP. Indeed, even if the reported average accuracy does not vary much across conditions, consistent rankings of the methods have been observed,

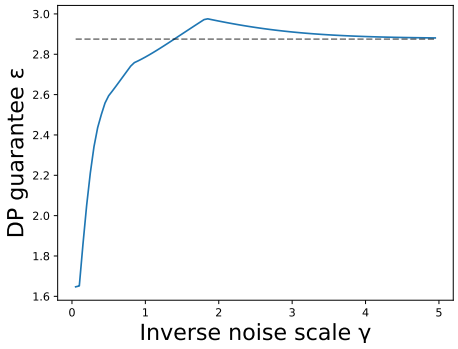

Figure 1.2: Differential privacy guarantees for MNIST as a function of γ, with $\tau = 0.9$

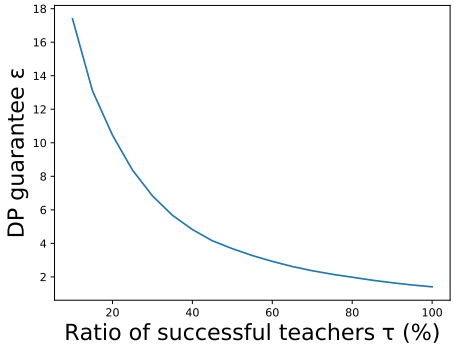

Figure 1.3: Differential privacy guarantees for MNIST as a function of τ, with $\gamma = 0.1$

confirming the expected average rank of the method based on the amount of added noise. As expected, the best DP guarantee ($\epsilon = 1.41$) is obtained when all the teachers generated noise ($\tau = 1$), but this is the case where the accuracy is the lowest. On the contrary, when some teachers failed to generate noise ($\tau = 0.9$ and $\tau = 0.7$), the counts are more precise, leading to a slightly better accuracy but worse DP guarantees. It should also be noted that the variance is high in each condition. It masks the fact that the distribution is highly skewed, with a majority of results in the $97.5\% - 98.5\%$ range, and a few samplings yielding an out-of-sample accuracy around 90%.

Table 1.2: Results for MNIST dataset with 250 teachers and 100 student queries. We used an inverse noise scale $\gamma = 0.1$. The DP guarantees, computed by composability with the moments accountant method over the 100 queries, are given for $\delta = 10^{-5}$.

Framework	ϵ	Acc. ($\pm$ std) [%]	HE overhead
Non-private	-	96.22 (± 2.27)	-
Trusted	1.41	95.95 (± 2.97)	-
$\tau = 1$	1.41	95.91 (± 2.57)	
$\tau = 0.9$	1.66	96.02 (± 2.92)	6.5 min
$\tau = 0.7$	2.37	96.06 (± 2.61)	

Figure 1.2 shows the evolution of our DP guarantee as a function of γ, with $\tau = 0.9$ fixed. Note that the privacy cost decreases for $\gamma \geq 2$ which may seem counter-intuitive but the reason is thoroughly explained in Section A.4 of the appendix. Anyway, we observed empirically that the privacy cost has a finite limit in $+\infty$ (approximately 2.87) and remains greater than this limit for any $\gamma \geq 2$. The asymptote is shown by a dashed line on Figure 1.2.

Figure 1.3 shows the evolution of the DP guarantee as a function of τ, with $\gamma = 0.1$ fixed. As explained before, the greater τ, the better the DP guarantee.

Performances on SVHN. Table 1.3 presents our experimental results on SVHN dataset [5]. The variance on the accuracy is much smaller than for MNIST dataset because the test set is constituted of 16032 samples. Similarly to the MNIST experiment, the accuracy and the privacy cost increase when less noise is applied because less teachers noised their votes (i.e. when τ is small). The DP guarantees are not as good as for MNIST, this is due to the high amount of queries (500) necessary to obtain a good accuracy because the learning task is more complex.

Table 1.3: SVHN experimental results for 500 queries, with noise inverse scale $\gamma = 0.1$, $\delta = 10^{-5}$

Framework	ϵ	Acc. [%]	HE overhead
Non-private	-	84.7	-
Trusted	4.73	83.7	-
$\tau = 1$	4.73	83.5	32.5 min
$\tau = 0.9$	5.59	83.8	
$\tau = 0.7$	8.16	84.6	

1.7 . Conclusion and open questions for further works

Our framework allows a group of agents to collaborate and put together their sensitive knowledge while protecting it via two complementary technologies - differential privacy and homomorphic encryption - against any entity contributing to the learning or having access to the final model. Crucially, our experiments showed that our method is practical for deep learning applications, combining high accuracy, mild computational overhead and privacy guarantees adapting to the number of malicious teachers.

An interesting further work could investigate the fault tolerance of the privacy guarantees with other noises (e.g. Gaussian noise) or other infinite divisions (Laplace distribution can also be infinitely divided using individual Gaussian noises or individual Laplace noises [81]). A more ambitious direction towards collaborative deep learning with privacy would be to design new aggregation operators, more suitable to FHE performances yet still providing good DP bounds. In particular, a linear or quadratic aggregation operator would be amenable to almost negligible homomorphic computations overhead. This lighter homomorphic layer would enable to extend the applicability of our framework to more complex datasets. Such aggregation operators would also allow to associate homomorphic calculations with verifiable computing techniques (e.g. [71]) whereby the server would provide an encrypted aggregation result along with a formal proof that aggregation was indeed done correctly. These perspectives would then allow to address threats beyond the honest-but-curious model.

[5] Note that our DP guarantee ϵ for Trusted cannot be directly compared with PATE's one since we do not use the same δ.

A - DP analysis of the learning procedure

In this appendix, we describe the procedure that computes the overall DP guarantees of the student model learning stage. We summarise this procedure in Section A.1, and demonstrate the theorems we use in Sections A.2 and A.4.

We call $\mathcal{A}$ the aggregation mechanism that outputs the argmax of the noisy counts. $\mathcal{A}(d, Q)$ is the output of $\mathcal{A}$ for the database d and the query Q.

Let $\gamma \in \mathbb{R}_+^*$ be the inverse scale parameter of the distributed noise. Considering the DP guarantees from the point of view of an entity $\mathcal{E}$, let $\tau \in (0, 1)$ be the ratio of the teachers whose noise is ignored by $\mathcal{E}$. Typically, from the point of view of a colluding teacher, τ is the ratio of the teachers who do not collude.

A.1 . Analysis algorithm

Let us suppose that for every query Q from the student model, we have a privacy guarantee using Theorem 4 and that we can upper bound the probability $\mathbb{P}[\mathcal{A}(d; Q) \neq k^*]$ that $\mathcal{A}$ outputs some specific output k^* (in practice we choose k^* to be the unnoisy argmax). Then, Theorem 5 from [141] gives us an upper bound on the moments accountant per query[1]. The computation of these building blocks is detailed in Sections A.2 and A.4, and the procedure is summarised in Algorithm 2.

Using the moments accountant per query, we evaluate the overall moments accountant by composability, applying Theorem 1 from [2]. Finally, parameter δ being chosen, the privacy guarantee is derived from the overall moments accountant applying the tail bound property, stated in Theorem 2 from [2].

[1]Note that only the third value over which the minimum is taken in Theorem 5 is data-dependent and, as such, requires this upper bound of $\mathbb{P}[\mathcal{A}(d; Q) \neq k^*]$.

Algorithm 2: Algorithm to determine the overall privacy guarantee of the learning procedure

Input : number of teachers n, number of classes K, ratio τ of teachers with secret noise, set of queries $\mathcal{Q}$, unnoisy teachers' counts n_k, inverse noise scale γ, l_{max} [a], δ

Output: ϵ

for l *in* $[l_{max}]$ **do**
  $\alpha(l) \leftarrow 0$
  for *query* Q *in* $\mathcal{Q}$ **do**
    Compute the privacy cost of Q and an upper bound of $\mathbb{P}[\mathcal{A}(d;Q) \neq k^*]$;
    Derive the moments accountant $\alpha_Q(l)$ with Theorem 5;
    $\alpha(l) \leftarrow \alpha(l) + \alpha_Q(l)$;
  end
  $\epsilon(l) \leftarrow \frac{\alpha(l)-\delta}{l}$;
end
$\epsilon \leftarrow \min_{l\in[l_{max}]} \epsilon(l)$;

[a] To determine the DP guarantees presented in Chapter 1, we took $l_{max} = 25$ because it seems empirically that it captures the best moments accountant in every case.

A.2 . DP guarantee per query in the BHBC framework

Preliminaries on the generalised Laplace distribution. For every teacher j who did send noise and whose noise is secret, the noise sent by j is distributed as $G_1^{(j)} - G_2^{(j)}$ where $G_1^{(j)}$ and $G_2^{(j)}$ are two i.i.d. random variables with gamma density $u \mapsto \frac{1}{\left(\frac{1}{\gamma}\right)^{\frac{1}{n}}\Gamma\left(\frac{1}{n}\right)} u^{\frac{1}{n}-1}e^{-\gamma u}$ and characteristic function $t \mapsto \left(\frac{1}{1-i\frac{t}{\gamma}}\right)^{\frac{1}{n}}$ (see [112]). Hence, the characteristic function of $G_1^{(j)} - G_2^{(j)}$ is $\psi\colon t \mapsto \left(\frac{1}{1+\left(\frac{t}{\gamma}\right)^2}\right)^{\frac{1}{n}}$. By summing over all the teachers who did send a secret noise, we get a total noise whose characteristic function is $\psi^{\tau n}\colon t \mapsto \left(\frac{1}{1+\left(\frac{t}{\gamma}\right)^2}\right)^{\tau}$. The corresponding moment generating function is $t \mapsto \left(\frac{1}{1-\left(\frac{t}{\gamma}\right)^2}\right)^{\tau}$. According to [125], this is the moment generating function of a generalised Laplace distribution whose density is

$$f_{\gamma,\tau}\colon u \in \mathbb{R}^* \mapsto \begin{cases} \frac{1}{\left(\frac{1}{\gamma}\right)^{2\tau}\Gamma(\tau)^2} e^{\gamma u} \int_u^{+\infty} t^{\tau-1}\left(t-u\right)^{\tau-1} e^{-2\gamma t} dt & \text{if} \quad u > 0 \\ \frac{1}{\left(\frac{1}{\gamma}\right)^{2\tau}\Gamma(\tau)^2} e^{\gamma u} \int_0^{+\infty} t^{\tau-1}\left(t-u\right)^{\tau-1} e^{-2\gamma t} dt & \text{if} \quad u < 0 \end{cases}$$

which is actually

$$
\begin{aligned}
u \in \mathbb{R}^* \mapsto & \frac{1}{\left(\frac{1}{\gamma}\right)^{2\tau} \Gamma(\tau)^2} e^{\gamma|u|} \int_{|u|}^{+\infty} t^{\tau-1} (t-|u|)^{\tau-1} e^{-2\gamma t} dt \\
&= \frac{\gamma^{2\tau-1}}{\Gamma(\tau)^2} e^{\gamma|u|} \int_0^{+\infty} \left(\frac{v}{\gamma} + |u|\right)^{\tau-1} \left(\frac{v}{\gamma}\right)^{\tau-1} e^{-2(v+\gamma|u|)} dv \\
&\qquad \text{(by the substitution } v = \gamma(t-|u|)) \\
&= L_{\gamma,\tau} e^{-\gamma|u|} I_\tau(\gamma|u|)
\end{aligned}
$$

where $I_\tau \colon v \in \mathbb{R}_+^* \mapsto \int_0^{+\infty} (x+v)^{\tau-1} x^{\tau-1} e^{-2x} dx$ and $L_{\gamma,\tau} = \frac{\gamma}{\Gamma(\tau)^2}$.

Let us remark that, since $\tau - 1 \leq 0$, I_τ is decreasing on $\mathbb{R}_+^*$.

As a density function, $f_{\gamma,\tau}$ is integrable on $\mathbb{R}$ (it can also be proved using Lemma 4). We call $F_{\gamma,\tau}$ the associated cumulative distribution function:

$$
F_{\gamma,\tau} \colon t \in \mathbb{R} \mapsto \int_{-\infty}^{t} f_{\gamma,\tau}(u) du
$$

Note that, $\lim_{+\infty} F_{\gamma,\tau} = 1$ and, since $f_{\gamma,\tau}$ is pair, $F_{\gamma,\tau}(0) = \frac{1}{2}$ and

$$
\forall t \in \mathbb{R}, F_{\gamma,\tau}(t) + F_{\gamma,\tau}(-t) = 1. \tag{A.1}
$$

If there is no ambiguity on the parameters γ and τ, we will only write f, F, I and L.

Lemma 1. *Let r be a random variable following the generalised Laplace distribution as defined above. Suppose that we can find a function M of γ and τ such that, for any $t \in \mathbb{R}$, $\frac{\mathbb{P}[r \geq t]}{\mathbb{P}[r \geq t+2]} \leq M$.*

Then $\mathcal{A}$ is $(\log(M), 0)$-differentially private.

Proof. We will mimic the proof of the privacy guarantee of the report noisy max from [66] (Claim 3.9), but with two key adaptations.

First of all, let us warn that our definition of the adjacency of two databases is different from the one of [66]. Changing one teacher is analogous to changing one individual in the *counting queries* context. This is why the hypotheses must be adapted. Indeed, d and d' being two adjacent databases (in our sense), since at most one teacher will change its vote between d and d', we have the property $|n_k - n'_k| \leq 1$ for any $k \in [K]$ but we do not have the property of monotonicity of the counts used in [66] [2].

The second difference is that, r being a random variable following the generalised Laplace distribution, we have to substitute the classical upper bound $e^{2\gamma}$ (valid for the Laplace distribution) of $\frac{\mathbb{P}[r \geq t]}{\mathbb{P}[r \geq t+2]}$ by M.

[2] We could have consider a database $\tilde{d}$ such that d is adjacent to $\tilde{d}$ and d' is adjacent to $\tilde{d}$ with Dwork's definition. Then we could have applied twice the result of [66] (using M instead of e^γ as upper bound of $\frac{\mathbb{P}[r \geq t]}{\mathbb{P}[r \geq t+2]}$ for $(d, \tilde{d})$ and $(\tilde{d}, d')$). Nevertheless, we performed numerical experimentations that make us believe that it would have given worse privacy guarantees than the present result.

We consider a query Q. Let $k_0 \in [K]$.

For any event E, we write $\mathbb{P}[E|r_{-k_0}]$ the probability of E under the condition that the draw from the $(K-1)$-dimensional generalised Laplace distribution, used for all the noisy counts except the k_0-th count, is equal to r_{-k_0}. We now suppose this draw r_{-k_0} fixed.

We define $r^* = \min\{r_{k_0} | \forall k \in [K] \setminus \{k_0\}, n_{k_0} + r_{k_0} \geq n_k + r_k\}$. Note that, whatever is the tie-breaking policy, r_{-k_0} being fixed, k_0 is the output of $\mathcal{A}$ for database d if $r_{k_0} > r^*$ and k_0 is not the output of $\mathcal{A}$ if $r_{k_0} < r^*$. Since $\mathbb{P}[r_{k_0} = r^*] = 0$, we have $\mathbb{P}[\mathcal{A}(d, Q) = k_0 | r_{-k_0}] = \mathbb{P}[r_{k_0} > r^*] = \mathbb{P}[r_{k_0} \geq r^*]$. Moreover, for all $k \in [K] \setminus \{k_0\}$,

$$\begin{aligned} n'_{k_0} + r^* + 2 &\geq n_{k_0} + r^* + 1 && \text{(because } |n_{k_0} - n'_{k_0}| \leq 1\text{)} \\ &\geq n_k + r_k + 1 && \text{(by definition of } r^*\text{)} \\ &\geq n'_k + r_k && \text{(because } |n_{k_0} - n'_{k_0}| \leq 1\text{)} \end{aligned}$$

We deduce that, if $r_{k_0} > r^* + 2$, then k_0 is the output of $\mathcal{A}$ for database d'. Therefore, $\mathbb{P}[\mathcal{A}(d', Q) = k_0 | r_{-k_0}] \geq \mathbb{P}[r_{k_0} > r^* + 2] = \mathbb{P}[r_{k_0} \geq r^* + 2]$.

Since $\mathbb{P}[r_{k_0} \geq r^*] \leq M\mathbb{P}[r_{k_0} \geq r^* + 2]$ by assumption, we can deduce that $\mathbb{P}[\mathcal{A}(d, Q) = k_0 | r_{-k_0}] \leq M\mathbb{P}[\mathcal{A}(d', Q) = k_0 | r_{-k_0}]$. This being true for any draw r_{-k_0}, the law of total probability gives us $\mathbb{P}[\mathcal{A}(d, Q) = k_0] \leq M\mathbb{P}[\mathcal{A}(d', Q) = k_0]$.

As d and d' play perfectly symmetric roles (unlike in the proof of the report noisy max guarantee from [66]), we also have $\mathbb{P}[\mathcal{A}(d', Q) = k_0] \leq M\mathbb{P}[\mathcal{A}(d, Q) = k_0]$. Since this is true for any query Q, we can conclude that $\mathcal{A}$ is $(\log(M), 0)$-differentially private. □

By definition of F, r being a random variable following the generalised Laplace distribution, for all $t \in \mathbb{R}$,

$$\mathbb{P}[r \geq t] = 1 - F(t).$$

Let $a \in \mathbb{R}_+^*$.

In the following, we exhibit upper bounds of $g \colon t \in \mathbb{R} \mapsto \frac{1-F(t)}{1-F(t+a)}$ (Propositions 6 and 7) to derive privacy guarantees for $\mathcal{A}$ (Theorem 4) taking $a = 2$. Let us first state some useful lemmas.

Lemma 2. *Let $\beta \in \mathbb{R}_+$. The application $h \colon z \in \mathbb{R}_+^* \mapsto \frac{I(z)}{I(z+\beta)}$ is decreasing.*

Proof. We will prove that h is differentiable and that its derivative is non-positive.

Let $\phi \colon (z, t) \in (\mathbb{R}_+^*)^2 \mapsto (t+z)^{\tau-1} t^{\tau-1} e^{-2\gamma t}$. ϕ has a partial derivative in the first variable and, for all $(z, t) \in (\mathbb{R}_+^*)^2$, $\frac{\partial \phi}{\partial z}(z, t) = (\tau - 1)(t+z)^{\tau-2} t^{\tau-1} e^{-2\gamma t}$. ϕ and $\frac{\partial \phi}{\partial z}$ are continuous in both variables.

Let $b \in \mathbb{R}_+^*$. For all $(z, t) \in [b, +\infty) \times \mathbb{R}_+^*$, $|\frac{\partial \phi}{\partial z}(z, t)| \leq \psi(t)$ where $\psi \colon t \in \mathbb{R}_+^* \mapsto (1-\tau)(t+b)^{\tau-2} t^{\tau-1} e^{-2\gamma t}$. ψ is continuous and integrable on $[b, +\infty)$. Applying Leibniz's theorem, we deduce that I is differentiable on $[b, +\infty)$ and that, for all $z \in [b, +\infty)$, $I'(z) = \int_0^{+\infty} (\tau - 1)(t+z)^{\tau-2} t^{\tau-1} e^{-2\gamma t} dt$. Since this is true for all $b \in \mathbb{R}_+^*$, we know that I is differentiable on $\mathbb{R}_+^*$ and that, for all $z \in \mathbb{R}_+^*$, $I'(z) = \int_0^{+\infty} (\tau-1)(t+z)^{\tau-2} t^{\tau-1} e^{-2\gamma t} dt$. As a consequence, h is differentiable on $\mathbb{R}_+^*$ and, for all $z \in \mathbb{R}_+^*$, $h'(z) = \frac{I(z+\beta)I'(z) - I(z)I'(z+\beta)}{I(z+\beta)^2}$.

Let $z \in \mathbb{R}_+^*$.

$$
\begin{aligned}
&I(z+\beta)I'(z) - I(z)I'(z+\beta) \\
&= \int_0^{+\infty} (x+z+\beta)^{\tau-1} x^{\tau-1} e^{-2x} dx \times \int_0^{+\infty} (\tau-1)(y+z)^{\tau-2} y^{\tau-1} e^{-2y} dy \\
&\quad - \int_0^{+\infty} (y+z)^{\tau-1} y^{\tau-1} e^{-2y} dy \times \int_0^{+\infty} (\tau-1)(x+z+\beta)^{\tau-2} x^{\tau-1} e^{-2x} dx \\
&= (\tau-1)\left[\int_0^{+\infty} (x+z+\beta)^{\tau-1} x^{\tau-1} e^{-2x} \int_0^{+\infty} (y+z)^{\tau-2} y^{\tau-1} e^{-2y} dy dx\right. \\
&\quad \left. - \int_0^{+\infty} (x+z+\beta)^{\tau-2} x^{\tau-1} e^{-2x} \int_0^{+\infty} (y+z)^{\tau-1} y^{\tau-1} e^{-2y} dy dx\right] \\
&= (\tau-1)\left[\int_0^{+\infty}\int_0^{+\infty} (x+z+\beta)^{\tau-1}(y+z)^{\tau-2}(xy)^{\tau-1} e^{-2(x+y)} dy dx\right. \\
&\quad \left. - \int_0^{+\infty}\int_0^{+\infty} (x+z+\beta)^{\tau-2}(y+z)^{\tau-1}(xy)^{\tau-1} e^{-2(x+y)} dy dx\right] \\
&= (\tau-1)\int_0^{+\infty}\int_0^{+\infty} (xy)^{\tau-1} e^{-2(x+y)} \\
&\qquad \times \left[(x+z+\beta)^{\tau-1}(y+z)^{\tau-2} - (x+z+\beta)^{\tau-2}(y+z)^{\tau-1}\right] dy dx \\
&= (\tau-1)\int_0^{+\infty}\int_0^{+\infty} (x+z+\beta)^{\tau-2}(y+z)^{\tau-2}(xy)^{\tau-1} e^{-2(x+y)} \\
&\qquad \times \left[(x+z+\beta) - (y+z)\right] dy dx \\
&= (\tau-1)\int_0^{+\infty}\int_0^{+\infty} (x+z+\beta)^{\tau-2}(y+z)^{\tau-2}(xy)^{\tau-1} e^{-2(x+y)} \\
&\qquad \times (x+\beta-y)\, dy dx \\
&\leq (\tau-1)\int_0^{+\infty}\int_0^{+\infty} (x+z+\beta)^{\tau-2}(y+z)^{\tau-2}(xy)^{\tau-1} e^{-2(x+y)}(x-y) dy dx \qquad \text{(A.2)}
\end{aligned}
$$

(because $\tau - 1 \leq 0$ and $\beta \geq 0$)

Similarly, we show that

$$
\begin{aligned}
&I(z+\beta)I'(z) - I(z)I'(z+\beta) \\
&= \int_0^{+\infty} (y+z+\beta)^{\tau-1} y^{\tau-1} e^{-2y} dy \times \int_0^{+\infty} (\tau-1)(x+z)^{\tau-2} x^{\tau-1} e^{-2x} dx \\
&\quad - \int_0^{+\infty} (x+z)^{\tau-1} x^{\tau-1} e^{-2x} dx \times \int_0^{+\infty} (\tau-1)(y+z+\beta)^{\tau-2} y^{\tau-1} e^{-2y} dy \\
&= (\tau-1)\left[\int_0^{+\infty} (x+z)^{\tau-2} x^{\tau-1} e^{-2x} \int_0^{+\infty} (y+z+\beta)^{\tau-1} y^{\tau-1} e^{-2y} dy dx \right. \\
&\quad \left. - \int_0^{+\infty} (x+z)^{\tau-1} x^{\tau-1} e^{-2x} \int_0^{+\infty} (y+z+\beta)^{\tau-2} y^{\tau-1} e^{-2y} dy dx \right] \\
&= (\tau-1) \int_0^{+\infty} \int_0^{+\infty} (xy)^{\tau-1} e^{-2(x+y)} \\
&\qquad \times \left[(x+z)^{\tau-2}(y+z+\beta)^{\tau-1} - (x+z)^{\tau-1}(y+z+\beta)^{\tau-2}\right] dydx \\
&= (\tau-1) \int_0^{+\infty} \int_0^{+\infty} (x+z)^{\tau-2}(y+z+\beta)^{\tau-2}(xy)^{\tau-1} e^{-2(x+y)} \\
&\qquad \times (y+\beta-x) dydx \\
&\leq (\tau-1) \int_0^{+\infty} \int_0^{+\infty} (x+z)^{\tau-2}(y+z+\beta)^{\tau-2}(xy)^{\tau-1} e^{-2(x+y)}(y-x) dydx \qquad \text{(A.3)}
\end{aligned}
$$

Alternatively, we can use A.2 to deduce A.3 directly using Fubini's theorem and exchanging the roles of x and y.

From A.2 and A.3, we get:

$$
\begin{aligned}
&2 \times \left[I(z+\beta)I'(z) - I(z)I'(z+\beta)\right] \\
&\leq (\tau-1) \int_0^{+\infty} \int_0^{+\infty} (x+z+\beta)^{\tau-2}(y+z)^{\tau-2}(x-y)(xy)^{\tau-1} e^{-2(x+y)} dydx \\
&\quad + (\tau-1) \int_0^{+\infty} \int_0^{+\infty} (x+z)^{\tau-2}(y+z+\beta)^{\tau-2}(y-x)(xy)^{\tau-1} e^{-2(x+y)} dydx \\
&= (\tau-1) \int_0^{+\infty} \int_0^{+\infty} (x-y)(xy)^{\tau-1} e^{-2(x+y)} \\
&\qquad \times \left[(x+z+\beta)^{\tau-2}(y+z)^{\tau-2} - (x+z)^{\tau-2}(y+z+\beta)^{\tau-2}\right] dydx
\end{aligned}
$$

Let $(x,y) \in (\mathbb{R}_+^*)^2$.

Note that $(x+z+\beta)(y+z) - (x+z)(y+z+\beta) = \beta(y-x)$ and then

$$
\begin{aligned}
&(x+z+\beta)^{\tau-2}(y+z)^{\tau-2} \geq (x+z)^{\tau-2}(y+z+\beta)^{\tau-2} \\
&\quad \Leftrightarrow (x+z+\beta)(y+z) \leq (x+z)(y+z+\beta) \qquad \text{(because } \tau-2<0\text{)} \\
&\quad \Leftrightarrow x \geq y.
\end{aligned}
$$

We deduce that

$$
\left[(x+z+\beta)^{\tau-2}(y+z)^{\tau-2} - (x+z)^{\tau-2}(y+z+\beta)^{\tau-2}\right](x-y) \geq 0.
$$

This inequality being true for all $(x,y) \in (\mathbb{R}_+^*)^2$ and, since $\tau - 1 \leq 0$, we have:

$$(\tau-1)\int_0^{+\infty}\int_0^{+\infty}\left[(x+z+\beta)^{\tau-2}(y+z)^{\tau-2}-(x+z)^{\tau-2}(y+z+\beta)^{\tau-2}\right] \times (x-y)(xy)^{\tau-1}e^{-2(x+y)}dydx \leq 0$$

Finally, $I(z+\beta)I'(z) - I(z)I'(z+\beta) \leq 0$ and $h'(z) \leq 0$.

Since this is true for any $z \in \mathbb{R}_+^*$, we can conclude that h is decreasing on $\mathbb{R}_+^*$. □

Lemma 3. *The function g has a maximum on $\mathbb{R}$, and this maximum is reached in the interval $[-\frac{a}{2}; 0]$.*

Proof. Since f is defined on $\mathbb{R}^*$, F is differentiable on $\mathbb{R}^*$. Thus g is differentiable on $\mathbb{R}^* \setminus \{-a\}$ and, for all $t \in \mathbb{R}^* \setminus \{-a\}$,

$$g'(t) = \frac{(1-F(t))f(t+a) - (1-F(t+a))f(t)}{(1-F(t+a))^2}.$$

First of all, let us prove that g is increasing on $(-\infty; -\frac{a}{2})$. For all $t \in (-\infty; -a)$, $|t| = -t \geq -t-a = |t+a|$ and, for all $t \in (-a; -\frac{a}{2})$, $|t| = -t \geq t+a = |t+a|$. Let $t \in (-\infty; -a) \cup (-a; -\frac{a}{2})$. Then, since $x \mapsto e^{-\gamma x}I(\gamma x)$ is decreasing on $\mathbb{R}_+^*$, $e^{-\gamma|t|}I(\gamma|t|) \leq e^{-\gamma|t+a|}I(\gamma|t+a|)$ which means $f(t) \leq f(t+a)$. Besides, F is increasing then, since $a \geq 0$, $1-F(t+a) \leq 1-F(t)$. Since $f(t)$, $f(t+a)$, $1-F(t)$ and $1-F(t)$ are all positive quantities, we deduce that $g'(t) \geq 0$. Then, g is increasing on $(-\infty; -a)$ and on $(-a; -\frac{a}{2})$ and since g is defined and continuous in $-a$, g is increasing on $(-\infty; -\frac{a}{2})$.

Let us now prove that g is decreasing on $\mathbb{R}_+$. Let $t \in \mathbb{R}_+^*$.

$$\begin{aligned}
&\frac{(1-F(t+a))^2}{L^2}g'(t)\\
&= \frac{1}{L^2}\left[(1-F(t))f(t+a) - (1-F(t+a))f(t)\right]\\
&= e^{-\gamma|t+a|}I(\gamma|t+a|)\int_t^{+\infty}e^{-\gamma|u|}I(\gamma|u|)du\\
&\quad - e^{-\gamma|t|}I(\gamma|t|)\int_{t+a}^{+\infty}e^{-\gamma|u|}I(\gamma|u|)du\\
&= e^{-\gamma(t+a)}I(\gamma(t+a))\int_t^{+\infty}e^{-\gamma u}I(\gamma u)du - e^{-\gamma t}I(\gamma t)\int_{t+a}^{+\infty}e^{-\gamma u}I(\gamma u)du\\
&= e^{-\gamma(t+a)}I(\gamma(t+a))\int_t^{+\infty}e^{-\gamma u}I(\gamma u)du\\
&\quad - e^{-\gamma t}I(\gamma t)\int_t^{+\infty}e^{-\gamma(v+a)}I(\gamma(v+a))dv\\
&\qquad \text{(by the substitution } v = u-a\text{)}\\
&= e^{-\gamma(t+a)}\left[\int_t^{+\infty}e^{-\gamma u}\left[I(\gamma(t+a))I(\gamma u) - I(\gamma t)I(\gamma(u+a))\right]du\right]
\end{aligned}$$

For any $u \in [t; +\infty)$, Lemma 2 with $\beta = \gamma a$ tells us that $\frac{I(\gamma u)}{I(\gamma(u+a))} \leq \frac{I(\gamma t)}{I(\gamma(t+a))}$ which means $I(\gamma(t+a))I(\gamma u) - I(\gamma t)I(\gamma(u+a)) \leq 0$.

Therefore, $\int_t^{+\infty} e^{-\gamma u} \left[I(\gamma(t+a))I(\gamma u) - I(\gamma t)I(\gamma(u+a))\right] du \leq 0$ and finally $g'(t) \leq 0$. This being valid for all $t \in \mathbb{R}_+^*$ and g being continuous in 0, we deduce that g is decreasing on $\mathbb{R}_+$.

From the two previous discussions and from the fact that g is continuous on $[-\frac{a}{2}; 0]$, we conclude that g has a maximum on $\mathbb{R}$ and that this maximum is reached in $[-\frac{a}{2}; 0]$. □

Proposition 6. *For all* $t \in [-\frac{a}{2}; 0]$*,*

$$g(t) \leq 1 + 2\frac{\int_0^{\frac{\gamma a}{2}} e^{-v} I(v) dv}{\int_{\gamma a}^{+\infty} e^{-v} I(v) dv}.$$

Proof. For all $t \in [-\frac{a}{2}; 0]$, $g(t) = 1 + \frac{F(t+a)-F(t)}{1-F(t+a)}$.

Calling $\phi \colon t \in [-\frac{a}{2}; 0] \mapsto F(t+a) - F(t)$, we know that ϕ is differentiable on $[-\frac{a}{2}; 0)$ and that $\phi' \colon t \in [-\frac{a}{2}; 0) \mapsto f(t+a) - f(t)$. Since $x \in \mathbb{R}_+^* \mapsto e^{-x} I(x)$ is decreasing, we have, for all $t \in [-\frac{a}{2}; 0)$,

$$\begin{aligned}
\phi'(t) \geq 0 &\Leftrightarrow e^{-\gamma|t+a|} I(\gamma|t+a|) \geq e^{-\gamma|t|} I(\gamma|t|) \\
&\Leftrightarrow |t+a| \leq |t| \\
&\Leftrightarrow t + a \leq -t && \text{(because } t + a \geq 0 \text{ and } t \leq 0\text{)} \\
&\Leftrightarrow t \leq -\frac{a}{2}
\end{aligned}$$

Since ϕ is continuous in 0, we deduce that ϕ is decreasing on $[-\frac{a}{2}; 0]$ and then, for all $t \in [-\frac{a}{2}; 0]$, $F(t+a) - F(t) \leq F(\frac{a}{2}) - F(-\frac{a}{2})$. Moreover, since F is increasing, for all $t \in [-\frac{a}{2}; 0]$, $1 - F(t+a) \geq 1 - F(a)$.

Finally, for all $t \in [-\frac{a}{2}; 0]$,

$$\begin{aligned}
g(t) &\leq 1 + \frac{F(\frac{a}{2}) - F(-\frac{a}{2})}{1 - F(a)} \\
&= 1 + \frac{L \int_{-\frac{a}{2}}^{\frac{a}{2}} e^{-\gamma|u|} I(\gamma|u|) du}{L \int_{a}^{+\infty} e^{-\gamma|u|} I(\gamma|u|) du} \\
&= 1 + \frac{\frac{L}{\gamma} \int_{-\frac{\gamma a}{2}}^{\frac{\gamma a}{2}} e^{-|v|} I(|v|) dv}{\frac{L}{\gamma} \int_{\gamma a}^{+\infty} e^{-|v|} I(|v|) dv} && \text{(by the substitutions } v = \gamma u\text{)} \\
&= 1 + \frac{\int_{-\frac{\gamma a}{2}}^{0} e^{-|v|} I(|v|) dv + \int_{0}^{\frac{\gamma a}{2}} e^{-|v|} I(|v|) dv}{\int_{\gamma a}^{+\infty} e^{-|v|} I(|v|) dv} \\
&= 1 + \frac{\int_{0}^{\frac{\gamma a}{2}} e^{-|v'|} I(|v'|) dv' + \int_{0}^{\frac{\gamma a}{2}} e^{-|v|} I(|v|) dv}{\int_{\gamma a}^{+\infty} e^{-|v|} I(|v|) dv} && \text{(by the substitution } v' = -v\text{)} \\
&= 1 + \frac{2 \int_{0}^{\frac{\gamma a}{2}} e^{-|v|} I(|v|) dv}{\int_{\gamma a}^{+\infty} e^{-|v|} I(|v|) dv} \\
&= 1 + 2 \frac{\int_{0}^{\frac{\gamma a}{2}} e^{-v} I(v) dv}{\int_{\gamma a}^{+\infty} e^{-v} I(v) dv}
\end{aligned}$$

□

Proposition 7. *Let us suppose that $\tau > \frac{1}{2}$.*

For all $t \in [-\frac{a}{2}; 0]$,

$$g(t) \leq g(0) - \frac{a}{2} g'(0).$$

with

$$g'(0) = \gamma \frac{\frac{\Gamma(\tau)^2}{2} e^{-\gamma a} I(\gamma a) - I(0) \int_{\gamma a}^{+\infty} e^{-v} I(v) dv}{\left(\int_{\gamma a}^{+\infty} e^{-v} I(v) dv \right)^2}.$$

Proof. The result basically comes from the fact that g is concave on $[\operatorname{argmax}(g); 0]$ which we prove hereafter.

From the proof of Lemma 3 we know that g is differentiable on $[-\frac{a}{2}; 0)$ and $g' \colon t \mapsto \frac{(1-F(t))f(t+a) - (1-F(t+a))f(t)}{(1-F(t+a))^2} = \frac{g(t)f(t+a) - f(t)}{1-F(t+a)}$. In the proof of Lemma 2, we saw that I is differentiable on $\mathbb{R}_+^*$ and thus f is differentiable on $\mathbb{R}_+^*$. Finally, we get that g' is differentiable

on $(-a;0)$ and, for all $t \in (-a;0)$,

$$\begin{aligned}
g''(t) &= \frac{1}{(1-F(t+a))^2}\big[(1-F(t+a))[g'(t)f(t+a)+g(t)f'(t+a)-f'(t)] \\
&\qquad +f(t+a)[g(t)f(t+a)-f(t)]\big] \\
&= \frac{1}{(1-F(t+a))^2}\big[(1-F(t+a))[g'(t)f(t+a)+g(t)f'(t+a)-f'(t)] \\
&\qquad +(1-F(t+a))f(t+a)g'(t)\big] \\
&= 2g'(t)\frac{f(t+a)}{1-F(t+a)} + \frac{(1-F(t+a))[g(t)f'(t+a)-f'(t)]}{(1-F(t+a))^2} \\
&= 2g'(t)\frac{f(t+a)}{1-F(t+a)} + \frac{(1-F(t))f'(t+a)-(1-F(t+a))f'(t)}{(1-F(t+a))^2}.
\end{aligned}$$

Since I' is strictly negative on $\mathbb{R}_+^*$, for all $u<0$, $f'(u) = L\gamma[e^{\gamma u}I(-\gamma u) - e^{\gamma u}I'(-\gamma u)] > 0$ and, for all $u>0$, $f'(u) = L\gamma[-e^{-\gamma u}I(\gamma u) + e^{-\gamma u}I'(\gamma u)] < 0$. Then, for all $t \in (-a;0)$, $f'(t)>0$ and $f'(t+a)<0$ and, since $1-F(t)>0$ and $1-F(t+a)>0$, $(1-F(t))f'(t+a)<0$ and $(1-F(t+a))f'(t)>0$. We deduce that, for all $t \in (-a;0)$,

$$g''(t) < 2g'(t)\frac{f(t+a)}{1-F(t+a)} + \frac{(1-F(t))f'(t+a)}{(1-F(t+a))^2} \tag{A.4}$$

where $2\frac{f(t+a)}{1-F(t+a)} > 0$ and $\frac{(1-F(t))f'(t+a)}{(1-F(t+a))^2} < 0$.

According to Lemma 3, g has a maximum, which is reached on $[-\frac{a}{2};0]$. Let $t_{max} = \operatorname{argmax}(g)$. If $t_{max} \neq 0$, we can argue that $g'(t_{max}) = 0$ and then, from Inequation A.4, g'' is strictly negative on a neighbourhood of t_{max}. This implies that g' is decreasing on a neighbourhood of $(t_{max})^+$ and then strictly negative on a neighbourhood of $(t_{max})^+$.

Removing the assumption that $t_{max} \neq 0$, we need to be slightly more subtle since g' is not differentiable in 0 (because I is not differentiable in 0).

Since $\tau > \frac{1}{2}$, $v \mapsto v^{2\tau-2}e^{-2v}$ is integrable on $\mathbb{R}_+^*$ and we can extend the definition of I to $\mathbb{R}_+$. This implies in particular that F and then g are differentiable on the whole interval $(-a;+\infty)$ (with $g'(0) = \frac{(1-F(0))f(a)-(1-F(a))f(0)}{(1-F(a))^2}$). Then $g'(t_{max}) = 0$ and, from Inequation A.4, $\lim\limits_{(t_{max})^+} g'' < \frac{(1-F(t_{max}))f'(t_{max}+a)}{(1-F(t_{max}+a))^2} < 0$. Thus g'' (not defined in 0) is strictly negative on a neighbourhood of $(t_{max})^+$. Then g' is strictly decreasing on a neighbourhood of $(t_{max})^+$ and, by continuity in t_{max}, strictly negative on a neighbourhood of $(t_{max})^+$.

Let us suppose that $g''(t) \geq 0$ for a t in $[t_{max};0)$ (trivially false if $t_{max}=0$ since $[t_{max};0)$ is empty in this case). We fix such a t and call it t_0. Then, from Inequation A.4, $g'(t_0) > 0$ and we can fix $t_1 = \inf\{t \in [t_{max};t_0] | g'(t) \geq 0\}$. g' is non-negative on a neighbourhood of $(t_1)^+$ thus $t_1 > t_{max}$. We also know that g' is non-positive on $[t_{max};t_1)$ by definition of t_1. This implies $g'(t_1) = 0$. Since $g'(t_1)=0$, from Inequation A.4, we know that $g''(t_1)<0$ and then g' is strictly negative on a neighbourhood of $(t_1)^+$. We get a contradiction so $g''(t)<0$ for all $t \in [t_{max};0)$. We deduce that g' is decreasing on $[t_{max};0)$.

Thus, for all $t \in [t_{max};0)$, $g'(t) \geq g'(0)$. As a consequence, since $t_{max} \leq 0$, $g(t_{max}) \leq g(0) + t_{max}g'(0)$. Besides, $t_{max} \geq -\frac{a}{2}$ and $g'(0) \leq g'(t_{max}) = 0$, thus $g(t_{max}) \leq g(0) - \frac{a}{2}g'(0)$.

Finally, by definition of t_{max}, for all $t \in \mathbb{R}$,

$$g(t) \leq g(0) - \frac{a}{2} g'(0)$$

with

$$\begin{aligned}
g'(0) &= \frac{(1-F(0))f(a) - (1-F(a))f(0)}{(1-F(a))^2} \\
&= \frac{\frac{1}{2}Le^{-\gamma a}I(\gamma a) - L^2 I(0)\int_a^{+\infty} e^{-\gamma u} I(\gamma u)du}{\left(L\int_a^{+\infty} e^{-\gamma u} I(\gamma u)du\right)^2} \\
&= \frac{\frac{1}{2L}e^{-\gamma a}I(\gamma a) - I(0)\int_a^{+\infty} e^{-\gamma u} I(\gamma u)du}{\left(\int_a^{+\infty} e^{-\gamma u} I(\gamma u)du\right)^2} \\
&= \frac{\frac{\Gamma(\tau)^2}{2\gamma}e^{-\gamma a}I(\gamma a) - \frac{1}{\gamma}I(0)\int_{\gamma a}^{+\infty} e^{-v} I(v)dv}{\left(\frac{1}{\gamma}\int_{\gamma a}^{+\infty} e^{-v} I(v)dv\right)^2} \qquad \text{(by the substitutions } v = \gamma u\text{)} \\
&= \gamma \frac{\frac{\Gamma(\tau)^2}{2}e^{-\gamma a}I(\gamma a) - I(0)\int_{\gamma a}^{+\infty} e^{-v} I(v)dv}{\left(\int_{\gamma a}^{+\infty} e^{-v} I(v)dv\right)^2}.
\end{aligned}$$

□

Theorem 6. *The aggregation mechanism $\mathcal{A}$ is $(\epsilon, 0)$-differentially private, with*

$$\epsilon = \log\left(1 + 2\frac{\int_0^{\gamma} e^{-v} I(v)dv}{\int_{2\gamma}^{+\infty} e^{-v} I(v)dv}\right).$$

Moreover, if $\tau > \frac{1}{2}$, g is differentiable in 0 and $\mathcal{A}$ is $(\epsilon', 0)$-differentially private, with

$$\epsilon' = \min\left[\epsilon, \log\left(g(0) - g'(0)\right)\right].$$

Proof. Thanks to Lemma 3, we can use Propositions 6 and 7 to upper bound g, for $a = 2$. We then just have to apply Lemma 1 to conclude. □

Lemma 4. *For all $v \in \mathbb{R}_+^*$, $I(v) \leq v^{\tau-1}\frac{\Gamma(\tau)}{2^\tau}$.*

Proof. Let $v \in \mathbb{R}_+^*$.

$$\begin{aligned}
I(v) &= \int_0^{+\infty} (t+v)^{\tau-1} t^{\tau-1} e^{-2t} dt \\
&\leq v^{\tau-1}\int_0^{+\infty} t^{\tau-1} e^{-2t} dt \qquad \text{(because } \tau - 1 \leq 0\text{)} \\
&= v^{\tau-1}\int_0^{+\infty} \left(\frac{u}{2}\right)^{\tau-1} e^{-u}\frac{du}{2} \qquad \text{(by the substitution } u = 2t\text{)} \\
&= v^{\tau-1}\frac{\Gamma(\tau)}{2^\tau}
\end{aligned}$$

□

Proposition 8. *For all* $\tau \in (0,1)$, $\lim_{\gamma \to 0} \left[\log \left(1 + 2 \frac{\int_0^\gamma e^{-v} I(v) dv}{\int_{2\gamma}^{+\infty} e^{-v} I(v) dv} \right) \right] = 0$.

Proof. For all $v \in \mathbb{R}_+^*$, $e^{-v} I(v) > 0$ thus, supposing $\gamma \in (0,1]$, $\int_{2\gamma}^{+\infty} e^{-v} I(v) dv \geq \int_2^{+\infty} e^{-v} I(v) dv > 0$. Therefore, it suffices to prove that $\lim_{\gamma \to 0} \left[\int_0^\gamma e^{-v} I(v) dv \right] = 0$ to deduce the announced result.

Applying Lemma 4, we get

$$\begin{aligned} \int_0^\gamma e^{-v} I(v) dv &\leq \frac{\Gamma(\tau)}{2^\tau} \int_0^\gamma e^{-v} v^{\tau-1} dv \\ &\leq \frac{\Gamma(\tau)}{2^\tau} \int_0^\gamma v^{\tau-1} dv \\ &= \frac{\Gamma(\tau)}{2^\tau} \frac{\gamma^\tau}{\tau} \end{aligned}$$

which gives $\lim_{\gamma \to 0} \left[\int_0^\gamma e^{-v} I(v) dv \right] = 0$. □

Proposition 9. *For all* $\gamma \in \mathbb{R}_+^*$, $\lim_{\tau \to 1} \left[\log \left(g(0) - g'(0) \right) \right] = 2\gamma$.

Proof. We use the dominated convergence theorem to determine the limit of f and F when τ approaches 1. Let us suppose in the following that $\tau \in (\frac{3}{4}; 1)$.

First of all, we determine the limit of I and deduce the one of f. Let $v \in \mathbb{R}_+$.

For all $x \in (0;1]$, $(x+v)^{\tau-1} x^{\tau-1} e^{-2x} \leq x^{2\tau-2} e^{-2x} \leq x^{-\frac{1}{2}} e^{-2x}$. As $x \mapsto x^{-\frac{1}{2}} e^{-2x}$ is integrable on $(0;1]$, and, for all $x \in (0;1]$,
$\lim_{\tau \to 1} \left[(x+v)^{\tau-1} x^{\tau-1} e^{-2x} \right] = e^{-2x}$, by the dominated convergence theorem we get that
$\lim_{\tau \to 1} \left[\int_0^1 (x+v)^{\tau-1} x^{\tau-1} e^{-2x} dx \right] = \int_0^1 e^{-2x} dx$.

Similarly, as, for all $x \in [1; +\infty)$, $(x+v)^{\tau-1} x^{\tau-1} e^{-2x} \leq e^{-2x}$ and
$\lim_{\tau \to 1} \left[(x+v)^{\tau-1} x^{\tau-1} e^{-2x} \right] = e^{-2x}$, by the dominated convergence theorem,
$\lim_{\tau \to 1} \left[\int_1^{+\infty} (x+v)^{\tau-1} x^{\tau-1} e^{-2x} dx \right] = \int_1^{+\infty} e^{-2x} dx$.

From the two points above, we deduce that

$$\begin{aligned} \lim_{\tau \to 1} I(v) &= \lim_{\tau \to 1} \left[\int_0^1 (x+v)^{\tau-1} x^{\tau-1} e^{-2x} dx + \int_1^{+\infty} (x+v)^{\tau-1} x^{\tau-1} e^{-2x} dx \right] \\ &= \int_0^1 e^{-2x} dx + \int_1^{+\infty} e^{-2x} dx \\ &= \int_0^{+\infty} e^{-2x} dx \\ &= \frac{1}{2} \end{aligned}$$

and, for any $u \in \mathbb{R}$, $\lim_{\tau \to 1} f(u) = \lim_{\tau \to 1} \left[\frac{\gamma}{\Gamma(\tau)^2} e^{-\gamma |u|} I(\gamma |u|) \right] = \frac{1}{2} \gamma e^{-\gamma |u|}$.

Let us now determine the limit of F.

Let $u_0 \in [0; \frac{1}{\gamma}]$ and $u_1 \in [0; \frac{1}{\gamma}]$ such that $u_0 < u_1$. According to Lemma 4, for all $u \in (u_0; u_1]$, $e^{-\gamma u} I(\gamma u) \leq e^{-\gamma u} (\gamma u)^{\tau-1} \frac{\Gamma(\tau)}{2^\tau} \leq e^{-\gamma u} (\gamma u)^{-\frac{1}{4}} \frac{\Gamma(\frac{3}{4})}{2^{\frac{3}{4}}}$ because $\gamma u \leq 1$ and Γ is decreasing on $(0; 1]$. Since $u \mapsto e^{-\gamma u} (\gamma u)^{-\frac{1}{4}} \frac{\Gamma(\frac{3}{4})}{2^{\frac{3}{4}}}$ is integrable on $(u_0; u_1]$ and, for all $u \in (u_0; u_1]$, $\lim\limits_{\tau \to 1} [e^{-\gamma u} I(\gamma u)] = \frac{e^{-\gamma u}}{2}$, by the dominated convergence theorem, $\lim\limits_{\tau \to 1} \left[\int_{u_0}^{u_1} e^{-\gamma u} I(\gamma u) du\right] = \int_{u_0}^{u_1} \frac{e^{-\gamma u}}{2} du$.

Let $u_0 \in [\frac{1}{\gamma}; +\infty)$ and $u_1 \in [\frac{1}{\gamma}; +\infty) \cup \{+\infty\}$ such that $u_0 < u_1$. Similarly, as, for all $u \in [u_0; u_1)$, $e^{-\gamma u} I(\gamma u) \leq e^{-\gamma u} (\gamma u)^{\tau-1} \frac{\Gamma(\tau)}{2^\tau} \leq e^{-\gamma u} \frac{\Gamma(\frac{3}{4})}{2^{\frac{3}{4}}}$. Since $u \mapsto e^{-\gamma u} \frac{\Gamma(\frac{3}{4})}{2^{\frac{3}{4}}}$ is integrable on $[u_0; u_1)$ and, for all $u \in [u_0; u_1)$, $\lim\limits_{\tau \to 1} [e^{-\gamma u} I(\gamma u)] = \frac{e^{-\gamma u}}{2}$, by the dominated convergence theorem,
$\lim\limits_{\tau \to 1} \left[\int_{u_0}^{u_1} e^{-\gamma u} I(\gamma u) du\right] = \int_{u_0}^{u_1} \frac{e^{-\gamma u}}{2} du$.

We deduce that, whatever are the bounds $u_0 \in [0; +\infty)$ and $u_1 \in [0; +\infty) \cup \{+\infty\}$ with $u_0 < u_1$, $\lim\limits_{\tau \to 1} \left[\int_{u_0}^{u_1} e^{-\gamma u} I(\gamma u) du\right] = \int_{u_0}^{u_1} \frac{e^{-\gamma u}}{2} du$. By substitution, we also have $\lim\limits_{\tau \to 1} \left[\int_{u_0}^{u_1} e^{\gamma u} I(-\gamma u) du\right] = \int_{u_0}^{u_1} \frac{e^{\gamma u}}{2} du$ for any $u_0 \in (-\infty; 0] \cup \{-\infty\}$ and $u_1 \in (-\infty; 0]$ with $u_0 < u_1$.

Finally, for any $u_0 \in (-\infty; 0] \cup \{-\infty\}$ and $u_1 \in [0; +\infty) \cup \{+\infty\}$ such that $u_0 < u_1$, we have $\lim\limits_{\tau \to 1} \left[\int_{u_0}^{u_1} e^{-\gamma|u|} I(\gamma|u|) du\right] = \int_{u_0}^{u_1} \frac{e^{-\gamma|u|}}{2} du$. In particular, for all $z \in \mathbb{R}$,

$$\begin{aligned}
\lim_{\tau \to 1} F(z) &= \lim_{\tau \to 1} (L) \times \int_{-\infty}^{z} \frac{e^{-\gamma|u|}}{2} du \\
&= \gamma \int_{-\infty}^{z} \frac{e^{-\gamma|u|}}{2} du \\
&= \begin{cases} \frac{1}{2} e^{\gamma z} \text{ if } z < 0 \\ 1 - \frac{1}{2} e^{-\gamma z} \text{ if } z \geq 0 \end{cases}
\end{aligned}$$

which is actually the expression of the Laplace cumulative distribution function.

From what precedes we can conclude that, with $a = 2$,

$$\begin{aligned}
&\lim_{\tau \to 1} \left[g(0) - g'(0)\right] \\
&\quad = \lim_{\tau \to 1} \left[\frac{1 - F(0)}{1 - F(2)} - \frac{(1 - F(0)) f(2) - (1 - F(2)) f(0)}{(1 - F(2))^2}\right] \\
&\quad = \frac{\frac{1}{2}}{\frac{1}{2} e^{-2\gamma}} - \frac{1}{2} \frac{\frac{1}{2} \times \frac{1}{2} \gamma e^{-2\gamma} - \frac{1}{2} e^{-2\gamma} \times \frac{1}{2} \gamma}{(\frac{1}{2} e^{-2\gamma})^2} \\
&\quad = e^{2\gamma}
\end{aligned}$$

□

A.3 . Influence of the HE layer on the DP guarantee per query

The computation of the homomorphic argmax induces some perturbations on the noisy counts and, as such, could harm the DP guarantees that we just gave. The three kinds of perturbations due to the HE layer are:

- the addition of (Gaussian) noise at the time of TFHE encryption which is inherently probabilistic
- the addition of a constant value A on the noisy counts to ensure that all the noisy counts are positive (with high probability) (see Section B)
- a possible mistake on the argmax if two noisy counts are too close (see Section 1.6)

While these perturbations can be seen as some post-processing applied on the clear noisy histogram, they cannot be seen as a post-processing on the clear noisy argmax on which we showed DP guarantees in Section A.2. Nevertheless, if we can prove that these perturbations consist of an addition of noise on the clear histogram, the upper bound on $\frac{\mathbb{P}[r \geq t]}{\mathbb{P}[r \geq t+2]}$, r being the total noise (generalised Laplace noise and HE perturbations) applied to the histogram of the n_k's, would still hold, leading to the same DP guarantees. The additions of Gaussian noise and constant A at encryption have, by commutativity, the same effect as the addition of a sum of Gaussian noises and nA after summation and they will anyway change the output of the homomorphic argmax with very low probability. However, some further work needs to be done in order to check whether the third kind of perturbation can be simulated as a noise addition on the histogram.

A.4 . Upper bound of the probability of a report noisy max mistake

In this subsection, we give an upper bound of the probability that $\mathcal{A}$ outputs a wrong argmax because of the added noise following the generalised Laplace distribution.

Lemma 5. *Let $u_0 \in \mathbb{R}_+$. Let $q \in \left(\frac{1}{1-\tau}; +\infty\right)$ and $p := \frac{1}{1-\frac{1}{q}}$.*

We have

$$\int_{u_0}^{+\infty} e^{-\gamma u} I(\gamma u) du \leq \frac{\Gamma(\tau)}{2^\tau \gamma} \frac{e^{-\gamma u_0}}{p^{\frac{1}{p}}} \frac{(\gamma u_0)^{\tau - 1 + \frac{1}{q}}}{[q(1-\tau)-1]^{\frac{1}{q}}}.$$

Proof. Let $u_0 \in \mathbb{R}_+$. Let $(p, q) \in \left(\mathbb{R}_+^*\right)^2$ such that $\frac{1}{p} + \frac{1}{q} = 1$ and $q > \frac{1}{1-\tau}$.

$$\begin{aligned}
&\int_{u_0}^{+\infty} e^{-\gamma u} I(\gamma u) du \\
&\quad \leq \frac{\Gamma(\tau)}{2^\tau} \int_{u_0}^{+\infty} e^{-\gamma u} (\gamma u)^{\tau - 1} du && \text{(according to Lemma 4)} \\
&\quad = \frac{\Gamma(\tau)}{2^\tau \gamma} \int_{\gamma u_0}^{+\infty} e^{-v} v^{\tau - 1} dv && \text{(by the substitution } v = \gamma u\text{)}
\end{aligned}$$

By assumption, $q > \frac{1}{1-\tau}$ so, since $\tau < 1$, $q(\tau - 1) < -1$ and then $v \in \mathbb{R}_+^* \mapsto v^{q(\tau-1)}$ is integrable in the neighbourhood of $+\infty$. Then we can apply Hölder's inequality in the

following manner:

$$\begin{aligned}
&\int_{u_0}^{+\infty} e^{-\gamma u} I(\gamma u) du \\
&\quad \leq \frac{\Gamma(\tau)}{2^\tau \gamma} \left(\int_{\gamma u_0}^{+\infty} e^{-pv} dv \right)^{\frac{1}{p}} \left(\int_{\gamma u_0}^{+\infty} v^{q(\tau-1)} dv \right)^{\frac{1}{q}} \\
&\quad = \frac{\Gamma(\tau)}{2^\tau \gamma} \times \left(\frac{e^{-p\gamma u_0}}{p} \right)^{\frac{1}{p}} \times \left(\frac{-(\gamma u_0)^{q(\tau-1)+1}}{(q(\tau-1)+1)} \right)^{\frac{1}{q}} \\
&\quad = \frac{\Gamma(\tau)}{2^\tau \gamma} \times \frac{e^{-\gamma u_0}}{p^{\frac{1}{p}}} \times \frac{(\gamma u_0)^{\tau - 1 + \frac{1}{q}}}{[q(1-\tau)-1]^{\frac{1}{q}}}
\end{aligned}$$

□

Lemma 6. *Let us consider a query Q. Let $k^* \in [K]$ be the unnoisy argmax (for all $k \in [K]$, $n_{k^*} \geq n_k$). For all $k \in [K]$, we define $\Delta_k := n_{k^*} - n_k \geq 0$. Then, for all $q \in (\frac{1}{1-\tau}; +\infty)$, calling $p := \frac{1}{1-\frac{1}{q}}$,*

$$\mathbb{P}[\mathcal{A}(d, Q) \neq k^*] \leq \sum_{k \neq k^*} e^{-\gamma \Delta_k} \left[\frac{1}{2} + \frac{1}{\tau 2^{4\tau - 2 + \frac{1}{q}} \Gamma(\tau)^2} \times \frac{(\gamma \Delta_k)^{2\tau - 1 + \frac{1}{q}}}{p^{\frac{1}{p}} [q(1-\tau)-1]^{\frac{1}{q}}} \right].$$

Proof. In the following, we will assume that $\Delta_k > 0$ and the upper bound for $\Delta_k = 0$ is obtained by continuity.

For any $k \in [K]$, let us denote Y_k the random variable following the generalised Laplace distribution generated by the sum of the τn individual noises.

Let $k \in [K]$.

$$\begin{aligned}
&\mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*}) \\
&\quad = \mathbb{P}(Y_{k^*} \leq Y_k - \Delta_k) \\
&\quad = \int_{-\infty}^{+\infty} f(t) F(t - \Delta_k) dt \\
&\quad = \int_{-\infty}^{0} f(t) F(t - \Delta_k) dt + \int_{0}^{\Delta_k} f(t) F(t - \Delta_k) dt + \int_{\Delta_k}^{+\infty} f(t) F(t - \Delta_k) dt \qquad \text{(A.5)}
\end{aligned}$$

We will now upper bound each one of the three above integrals separately. The two

extreme integrals can be nicely bounded by decreasing exponentials in Δ_k:

$$
\begin{aligned}
&\int_{\Delta_k}^{+\infty} f(t)F(t-\Delta_k)dt \\
&= \int_0^{+\infty} f(v+\Delta_k)F(v)dv && \text{(by the substitution } v = t-\Delta_k\text{)} \\
&= L\int_0^{+\infty} e^{-\gamma|v+\Delta_k|} I(\gamma|v+\Delta_k|)F(v)dv \\
&= L\int_0^{+\infty} e^{-\gamma(v+\Delta_k)} I(\gamma(v+\Delta_k))F(v)dv \\
&= Le^{-\gamma\Delta_k}\int_0^{+\infty} e^{-\gamma v} I(\gamma(v+\Delta_k))F(v)dv \\
&\leq Le^{-\gamma\Delta_k}\int_0^{+\infty} e^{-\gamma v} I(\gamma v)F(v)dv && \text{(because } I \text{ is decreasing)} \\
&= Le^{-\gamma\Delta_k}\int_0^{+\infty} e^{-\gamma|v|} I(\gamma|v|)F(v)dv \\
&= e^{-\gamma\Delta_k}\int_0^{+\infty} f(v)F(v)dv \\
&= e^{-\gamma\Delta_k} \times \frac{\lim^{+\infty} F^2 - F(0)^2}{2} \\
&= e^{-\gamma\Delta_k} \times \frac{1-\frac{1}{4}}{2} \\
&= \frac{3}{8}e^{-\gamma\Delta_k} && \text{(A.6)}
\end{aligned}
$$

and

$$
\begin{aligned}
&\int_{-\infty}^{0} f(t)F(t-\Delta_k)dt \\
&= L\int_{-\infty}^{0} f(t)\int_{-\infty}^{t-\Delta_k} e^{-\gamma|u|}I(\gamma|u|)dudt \\
&= L\int_{-\infty}^{0} f(t)\int_{-\infty}^{t-\Delta_k} e^{\gamma u}I(-\gamma u)dudt \\
&= L\int_{-\infty}^{0} f(t)\int_{-\infty}^{t} e^{\gamma(v-\Delta_k)}I(\gamma(\Delta_k-v))dudt \\
&\qquad\qquad \text{(by the substitution } v=u+\Delta_k\text{)} \\
&= Le^{-\gamma\Delta_k}\int_{-\infty}^{0} f(t)\int_{-\infty}^{t} e^{\gamma v}I(\gamma(\Delta_k-v))dudt \\
&\leq Le^{-\gamma\Delta_k}\int_{-\infty}^{0} f(t)\int_{-\infty}^{t} e^{\gamma v}I(-\gamma v)dudt \qquad \text{(because } I \text{ is decreasing)} \\
&= Le^{-\gamma\Delta_k}\int_{-\infty}^{0} f(t)\int_{-\infty}^{t} e^{-\gamma|v|}I(\gamma|v|)dudt \\
&= e^{-\gamma\Delta_k}\int_{-\infty}^{0} f(t)F(t)dt \\
&= e^{-\gamma\Delta_k}\times\frac{F(0)^2-\lim_{-\infty}F^2}{2} \\
&= \frac{1}{8}e^{-\gamma\Delta_k}. \qquad\qquad (A.7)
\end{aligned}
$$

As for the middle integral, we have

$$
\begin{aligned}
&\int_{0}^{\Delta_k} f(t)F(t-\Delta_k)dt \\
&= L\int_{0}^{\Delta_k} f(t)\int_{-\infty}^{t-\Delta_k} e^{-\gamma|u|}I(\gamma|u|)dudt \\
&= L\int_{0}^{\Delta_k} f(t)\int_{\Delta_k-t}^{+\infty} e^{-\gamma|v|}I(\gamma|v|)dvdt \qquad \text{(by the substitution } v=-u\text{)} \\
&= L\int_{0}^{\Delta_k} f(t)\int_{\Delta_k-t}^{+\infty} e^{-\gamma v}I(\gamma v)dvdt
\end{aligned}
$$

Since, for all $t\in[0;\Delta_k]$, $0\leq\Delta_k-t$, we can apply Lemma 5. Let $q\in\left(\frac{1}{1-\tau};+\infty\right)$ and $p=\frac{1}{1-\frac{1}{q}}$. We have, for all $t\in(0;\Delta_k)$, $\int_{\Delta_k-t}^{+\infty}e^{-\gamma v}I(\gamma v)dv\leq\frac{\Gamma(\tau)}{2^\tau\gamma}\times\frac{1}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}}\times e^{-\gamma(\Delta_k-t)}[\gamma(\Delta_k-t)]^{\tau-1+\frac{1}{q}}$. Since $\tau-1+\frac{1}{q}>-1$, $t\mapsto[\gamma(\Delta_k-t)]^{\tau-1+\frac{1}{q}}$ is integrable on a neighbourhood of $(\Delta_k)^-$ and then, since $t\mapsto f(t)e^{-\gamma(\Delta_k-t)}$ is bounded on a neigh-

bourhood of Δ_k, $t \mapsto f(t)e^{-\gamma(\Delta_k - t)}[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}$ is integrable on a neighbourhood of $(\Delta_k)^-$.

Thus, we can write

$$\begin{aligned}
&\int_0^{\Delta_k} f(t)F(t-\Delta_k)dt \\
&\quad \leq L\frac{\Gamma(\tau)}{2^\tau\gamma} \times \frac{1}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \times \int_0^{\Delta_k} f(t)e^{-\gamma(\Delta_k - t)}[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}dt \\
&\quad = L^2\frac{\Gamma(\tau)}{2^\tau\gamma} \times \frac{1}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \\
&\qquad\qquad \times \int_0^{\Delta_k} e^{-\gamma|t|}I(\gamma|t|)e^{-\gamma(\Delta_k - t)}[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}dt \\
&\quad = \frac{\gamma}{2^\tau\Gamma(\tau)^3} \times \frac{1}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \\
&\qquad\qquad \times \int_0^{\Delta_k} e^{-\gamma t}I(\gamma t)e^{-\gamma(\Delta_k - t)}[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}dt \\
&\quad = \frac{e^{-\gamma\Delta_k}}{2^\tau\Gamma(\tau)^3} \times \frac{\gamma}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \times \int_0^{\Delta_k} I(\gamma t)[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}dt
\end{aligned}$$

$t \mapsto (\gamma t)^{\tau-1}$ is integrable on a neighbourhood of 0^+ because $\tau - 1 > -1$. Therefore, $t \mapsto (\gamma t)^{\tau-1}\frac{\Gamma(\tau)}{2^\tau}[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}$ is integrable on $(0;\Delta_k)$ so we can apply Lemma 4:

$$\begin{aligned}
&\int_0^{\Delta_k} f(t)F(t-\Delta_k)dt \\
&\quad \leq \frac{e^{-\gamma\Delta_k}}{2^\tau\Gamma(\tau)^3} \times \frac{\gamma}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \times \int_0^{\Delta_k} (\gamma t)^{\tau-1}\frac{\Gamma(\tau)}{2^\tau}[\gamma(\Delta_k - t)]^{\tau-1+\frac{1}{q}}dt \\
&\quad = \frac{e^{-\gamma\Delta_k}}{2^{2\tau}\Gamma(\tau)^2} \times \frac{\gamma\Delta_k}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \times \int_0^1 (\gamma\Delta_k u)^{\tau-1}[\gamma(\Delta_k - \Delta_k u)]^{\tau-1+\frac{1}{q}}du \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(by the substitution } u = \frac{t}{\Delta_k}\text{)} \\
&\quad = \frac{e^{-\gamma\Delta_k}}{2^{2\tau}\Gamma(\tau)^2} \times \frac{(\gamma\Delta_k)^{2\tau-1+\frac{1}{q}}}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}} \times \int_0^1 u^{\tau-1}(1-u)^{\tau-1+\frac{1}{q}}du
\end{aligned}$$

Note that

$$
\begin{aligned}
&\int_0^1 u^{\tau-1}(1-u)^{\tau-1+\frac{1}{q}}du \\
&= \int_0^{\frac{1}{2}} u^{\tau-1}(1-u)^{\tau-1+\frac{1}{q}}du + \int_{\frac{1}{2}}^1 u^{\tau-1}(1-u)^{\tau-1+\frac{1}{q}}du \\
&\leq \int_0^{\frac{1}{2}} u^{\tau-1}\frac{1}{2^{\tau-1+\frac{1}{q}}}du + \int_{\frac{1}{2}}^1 \frac{1}{2^{\tau-1}}(1-u)^{\tau-1+\frac{1}{q}}du \\
&\qquad\qquad \text{(because } \tau-1+\frac{1}{q}<0 \text{ and } \tau-1<0\text{)} \\
&= \frac{1}{2^{\tau-1+\frac{1}{q}}}\int_0^{\frac{1}{2}} u^{\tau-1}du + \frac{1}{2^{\tau-1}}\int_0^{\frac{1}{2}} v^{\tau-1+\frac{1}{q}}dv \\
&\qquad\qquad \text{(by the substitution } v=1-u\text{)} \\
&= \frac{1}{2^{\tau-1+\frac{1}{q}}}\times\frac{1}{\tau 2^{\tau}} + \frac{1}{2^{\tau-1}}\times\frac{1}{(\tau+\frac{1}{q})2^{\tau+\frac{1}{q}}} \\
&= \frac{1}{2^{2\tau-1+\frac{1}{q}}}\left(\frac{1}{\tau}+\frac{1}{\tau+\frac{1}{q}}\right) \\
&\leq \frac{1}{\tau 2^{2\tau-2+\frac{1}{q}}}
\end{aligned}
$$

Therefore

$$\int_0^{\Delta_k} f(t)F(t-\Delta_k)dt \leq \frac{e^{-\gamma\Delta_k}}{\tau 2^{4\tau-2+\frac{1}{q}}\Gamma(\tau)^2}\times\frac{(\gamma\Delta_k)^{2\tau-1+\frac{1}{q}}}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}}. \tag{A.8}$$

Using A.5, A.6, A.7 and A.8, we get

$$\mathbb{P}(n_k+Y_k\geq n_{k^*}+Y_{k^*}) \leq e^{-\gamma\Delta_k}\left[\frac{1}{2}+\frac{1}{\tau 2^{4\tau-2+\frac{1}{q}}\Gamma(\tau)^2}\times\frac{(\gamma\Delta_k)^{2\tau-1+\frac{1}{q}}}{p^{\frac{1}{p}}[q(1-\tau)-1]^{\frac{1}{q}}}\right].$$

The overall upper bound for $\mathbb{P}[\mathcal{A}(d;Q)\neq k^*]$ is obtained using the fact that the event $(\mathcal{A}(d;Q)\neq k^*)$ is the union of the events $(n_k+Y_k\geq n_{k^*}+Y_{k^*})$, for $k\in[K]\setminus\{k^*\}$, and then $\mathbb{P}[\mathcal{A}(d;Q)\neq k^*]\leq\sum_{k\neq k^*}\mathbb{P}(n_k+Y_k\geq n_{k^*}+Y_{k^*})$. □

Proposition 10. *If* $\tau\in(\frac{1}{2};1)$*,*

$$\mathbb{P}[\mathcal{A}(d;Q)\neq k^*] \leq \sum_{k\neq k^*} e^{-\gamma\Delta_k}\left[\frac{1}{2}+\frac{(\gamma\Delta_k)^{2\tau-1}}{\tau 2^{4\tau-2}\Gamma(\tau)^2}\right].$$

If $\tau\in(0;\frac{1}{2}]$*,*

$$\mathbb{P}[\mathcal{A}(d;Q)\neq k^*] \leq \sum_{k\neq k^*} e^{-\gamma\Delta_k}\left[\frac{1}{2}+\frac{(\gamma\Delta_k)^{\frac{\tau}{2}}}{\tau 2^{\frac{5}{2}\tau-1}\Gamma(\tau)^2}\times\left(\frac{3}{2}\tau\right)^{\frac{3}{2}\tau}\left(\frac{2}{\tau}-3\right)^{1-\frac{3}{2}\tau}\right].$$

Proof. Let us distinct two cases according to the value of τ.

First case: $\tau > \frac{1}{2}$

Taking the limit when q approaches $+\infty$ in A.8 (which actually amounts to substitute $v^{\tau-1}$ by its upper bound $(\gamma u_0)^{\tau-1}$ in the integral $\int_{\gamma u_0}^{+\infty} e^{-v} v^{\tau-1} dv$ of the proof of Lemma 5, without needing Hölder's inequality), we get

$$\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \leq \sum_{k \neq k^*} e^{-\gamma \Delta_k} \left[\frac{1}{2} + \frac{(\gamma \Delta_k)^{2\tau-1}}{\tau 2^{4\tau-2} \Gamma(\tau)^2} \right]$$

Second case: $\tau \leq \frac{1}{2}$

By convention, if $\tau = \frac{1}{2}$, we have $\frac{1}{1-2\tau} = +\infty$.

We take $q < \frac{1}{1-2\tau}$ (it is possible since $\frac{1}{1-2\tau} > \frac{1}{1-\tau}$) and write $q = \frac{1}{1-2\tau+\epsilon}$, with $0 < \epsilon < \tau$. Then, $\frac{1}{p} = 1 - \frac{1}{q} = 2\tau - \epsilon$ and we get

$$\begin{aligned} &\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \\ &\quad \leq \sum_{k \neq k^*} e^{-\gamma \Delta_k} \left[\frac{1}{2} + \frac{(2\tau-\epsilon)^{2\tau-\epsilon}}{\tau 2^{2\tau-1+\epsilon} \Gamma(\tau)^2} \times \left(\frac{1-2\tau+\epsilon}{\tau-\epsilon} \right)^{1-2\tau+\epsilon} \times (\gamma \Delta_k)^{\epsilon} \right] \end{aligned}$$

For example, with $\epsilon = \frac{\tau}{2}$ (i.e. $q = \frac{1}{1-\frac{3}{2}\tau}$), we have

$$\begin{aligned} &\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \\ &\quad \leq \sum_{k \neq k^*} e^{-\gamma \Delta_k} \left[\frac{1}{2} + \frac{1}{\tau 2^{\frac{5}{2}\tau-1} \Gamma(\tau)^2} \times \left(\frac{3}{2}\tau \right)^{\frac{3}{2}\tau} \left(\frac{2}{\tau} - 3 \right)^{1-\frac{3}{2}\tau} \times (\gamma \Delta_k)^{\frac{\tau}{2}} \right] \end{aligned}$$

□

Note that, whatever is the value of $\tau \in (0;1)$, our upper bound of $\mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*})$ tends to 0 when Δ_k approaches $+\infty$ which follows the intuition that $\mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*})$ tends to 0 when the true argmax k^* has a much higher count than k. The upper bound tends to $\frac{1}{2}$ when Δ_k approaches 0, which is consistent with the actual value of the probability $\mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*})$ when the counts n_{k^*} and n_k are equal.

Similarly, the upper bound tends to 0 when γ tends to $+\infty$ and to $\frac{1}{2}$ when γ approaches 0. These are the expected values of the probability $\mathbb{P}(n_k + Y_k \geq n_{k^*} + Y_{k^*})$ when there is no noise or an infinitely wide noise respectively.

Finally, let us remark that we recover the upper bound $\mathbb{P}[\mathcal{A}(d;Q) \neq k^*] \leq \sum_{k \neq k^*} \frac{2+\gamma\Delta_k}{4e^{\gamma\Delta_k}}$ from [141] (obtained with a centralised Laplace noise) when we consider the limit when τ tends to 1.

Remark. The data-dependent bound $\alpha_{\mathcal{A}}(l; aux, d, d') \leq \log\left((1-q)\left(\frac{1-q}{1-e^{\epsilon}q}\right)^{l} + qe^{\epsilon l}\right)$ from Theorem 5 is non-monotonic in γ. This may appear counter-intuitive since a smaller noise (greater γ) usually gives worse privacy guarantees and, as one would expect, a bigger moments accountant. Nevertheless, a smaller noise means that the probability of outputting the true

(unnoisy) argmax is closer to 1, which may lower the moments accountant. Indeed, two adjacent databases will both output the true argmax with high probability, giving less chance to an adversary to distinguish them. This non-monotonicity of the data-dependent bound induces the non-monotonicity of the overall privacy cost ϵ. This is illustrated in Figure 1.2 on which we can see, however, that choosing a small γ still gives better guarantees.

B - FHE argmax implementation details

We implemented the FHE argmax algorithm using the C++ TFHE library [51]. Table B.1 presents all of the parameters needed to reproduce our results and build a fully homomorphic argmax scheme using the TFHE library. The first two lines present our values for the standard TFHE parameters: the first line for initial ciphertext encryption; the second line for the two bootstrapping keys we use. Given the parameters that we use here, we achieve a security parameter of 110. We base the security of our scheme on the `lwe-estimator`[1] script. The estimator is based on the work presented in [8] and is consistently kept up to date.

Table B.1: Parameters for our implementation. The top line presents the overall security (λ), and the parameters for the initial encryption: σ is the Gaussian noise parameter and N is the size of polynomials. In the TFHE encryption scheme, there is a parameter k (different from the one used in Chapter 1) which, in our case, is always equal to 1. The second line presents the parameters needed to create the two bootstrapping keys we are using. For these two lines, we used the notations from [197] and [50]. The third line presents parameters specific to our implementation given the specificities of the data to process. A is the value to add to the ciphertexts before subtracting $n_k + Y_k - n_{k'} - Y_{k'}$ as per the notations in Section 1.4.3. b_i is the modulus with which the values are rescaled at encryption time to obtain values in $[0, 1]$ and to allow for a correct result of the θ computation. $b_\theta^{(1)}$ is the output modulus of the first bootstrapping operation creating the θ values. $b_\theta^{(2)}$ is the output modulus of the second and final bootstrapping operation.

N	σ		
1024	$1\mathsf{e}{-9}$		
N_b	σ_b	B_g	ℓ
1024	$1e$-9	64	6
A	b_i	$b_\theta^{(1)}$	$b_\theta^{(2)}$
900	4102	36	4

The third line presents parameters that are specific to our implementation. Because of the use of Gamma distributions, the values sent by the teachers can be negative. This can be an important issue: if a value is negative, then it will be interpreted in the ciphertext space as a very high positive value and the resulting argmax will be wrong. Therefore, after summing the ciphertexts from the teachers, we add a constant value (we can add a clear value to a ciphertext value) A to ensure that the $n_k + Y_k + A$ are all positive before subtraction. We evaluated that, given the parameters of the Gamma distributions used, choosing $A = 900$ gives us less than a 2^{-64} probability of failure: with Y_k following a Laplace distribution (as seen in Section 1.4), then we have $\mathbb{P}(Y_k < -A) < 2^{-64}$. The b_i variable corresponds to the value by which we rescale the cleartexts before encryption. Indeed, the cleartext and ciphertext spaces of the TFHE encryption scheme are both $\mathbb{T} = ([0, 1], +)$. Additionally, for a correct θ computation, we need to have $|\frac{n_k+Y_k-n_{k'}-Y_{k'}}{b_i}| < \frac{1}{2}$, which is true if, for all $k \in [K]$, $\frac{n_k+Y_k+A}{b_i} \in [0, \frac{1}{2})$. Since

[1] https://bitbucket.org/malb/lwe-estimator/raw/HEAD/estimator.py

$\mathbb{P}(Y_k \geq A) < 2^{-64}$ by symmetry, $b_i = 2(n + 2A) = 4100$ (with n the number of teachers) is sufficient to have $|\frac{n_k + Y_k - n_{k'} - Y_{k'}}{b_i}| < \frac{1}{2}$ with high probability. $b_\theta^{(1)}$ is the output modulus of the first bootstrapping operation. It needs to be chosen so that we have $\Theta_k > \frac{1}{2}$ for one and only one k. That k will then be considered the argmax. $b_\theta^{(2)}$ is the modulus for the final bootstrapping operation.

C - Detailed experimental settings

In this section, we provide the reader with additional details regarding experimental settings. In order to reproduce experimental results, all necessary source codes are available on https://github.com/Arnaud-GS/SPEED.

C.1 . Experimental settings for MNIST

Following PATE experimental conditions, we built our framework based on the code repositories[1] accompanying [141]. The teacher models are based on two convolutional layers with max-pooling and one fully connected layer with ReLUs. Code modifications have been performed on the initial repository, and are available on https://github.com/Arnaud-GS/SPEED. The execution environment consists in Python 3 and Tensorflow 1.15.0. The batch size, learning rate and max steps parameters have been respectively set to 128, 0.01 and 5000. As stated in [141], this yields an aggregate test-error rate of 93%. A semi-supervised technique proposed in [159] has been used[2], in an execution environment consisting of Python 3 and Theano 0.7. Besides modifications available on https://github.com/Arnaud-GS/SPEED, the learning rate and number of epochs have been set to 0.001 and 500 respectively.

C.2 . Experimental settings for SVHN

For SVHN, two additional layers have been added to the teacher models which were learned using a node with 8 NVIDIA v100. The batch size, learning rate and max steps parameters have been respectively set to 64, 0.08 and 2000. The student model also uses the improved GAN semi-supervised model, relying on Python 3 and Theano 0.8.2. The learning rate and number of epochs have been set to 0.0003 and 600 respectively.

[1] https://github.com/tensorflow/privacy/tree/master/research/pate_2017
[2] https://github.com/openai/improved-gan

2 - When approximate design for fast homomorphic computation provides differential privacy guarantees

Abstract While machine learning has become pervasive in as diversified fields as industry, healthcare, social networks, privacy concerns regarding the training data have gained a critical importance. In settings where several parties wish to collaboratively train a common model without jeopardising their sensitive data, the need for a private training protocol is particularly stringent and implies to protect the data against both the model's end-users and the actors of the training phase. Differential privacy (DP) and cryptographic primitives are complementary popular countermeasures against privacy attacks. Among these cryptographic primitives, fully homomorphic encryption (FHE) offers ciphertext malleability at the cost of time-consuming operations in the homomorphic domain. In this paper, we design SHIELD, a probabilistic approximation algorithm for the argmax operator which is both fast when homomorphically executed and whose inaccuracy is used as a feature to ensure DP guarantees. Even if SHIELD could have other applications, we here focus on one setting and seamlessly integrate it in the SPEED collaborative training framework from [83] to improve its computational efficiency. After thoroughly describing the FHE implementation of our algorithm and its DP analysis, we present experimental results. To the best of our knowledge, it is the first work in which relaxing the accuracy of the algorithm is constructively usable as a degree of freedom to achieve better FHE performances.

N.B.: This chapter is the reproduction of the article *When lightweight design for fast homomorphic computation provides differential privacy guarantees*, joint work with Martin Zuber, Oana Stan, Renaud Sirdey and Cédric Gouy-Pailler, to be submitted [85].

2.1 . Introduction

As a protocol for training neural network without explicit sharing of the learning data, the Private Aggregation of Teacher Ensembles (PATE) approach has received much attention since its inception in the seminal work of Papernot et al [141]. In a nutshell, the PATE protocol labels a subset of a public dataset and uses this partially labeled dataset to train a student model in a semi-supervised way. The labelization is achieved by aggregating, usually by means of majority voting, the labels - considered as votes - provided by a set of teachers which are the owners of private data sets. Since the teachers' labels would leak information on their training data, the PATE protocol makes use of differential privacy (DP). To get a reasonable privacy-utility trade-off, the vote aggregation is performed on an independent server, the single elected vote seen by the student model being much easier to sanitize than the full histogram of the votes.

Still, in such a setting, the server has to be trusted since it sees the clear votes sent by the teachers. This is why SPEED [83] builds upon the work from [141] and uses fully homomorphic

encryption (FHE) to blind the server by having it performing the aggregation directly over encrypted votes, therefore with neither knowledge of the individual votes nor of the consolidated one. In that work the authors associate a distributed Laplacian noise generation mechanism and carefully crafted homomorphic histogram and argmax computations. Still, FHE being computationally intensive, this comes at significant communication and computation costs on the server (6.5 minutes to compute the homomorphic argmax for 100 queries).

In this paper, we revisit the association of DP and FHE in a radically different fashion. Indeed, rather than proceeding in two steps (noise addition and then homomorphic aggregation) we proceed by designing a new aggregation algorithm which has the desirable property of being much more efficient to evaluate over FHE but the less desirable property of being (stochastically) inaccurate. We then demonstrate that the inaccuracies of our algorithm translate into consistent DP guarantees, and therefore that explicit noise addition becomes unnecessary for DP. In doing so, and by means of a carefully crafted FHE implementation of the algorithm, we are able to achieve a reduction of 20% in the computational burden of the aggregation server compared to the state of the art [97]. By opposition to CKKS-based approaches in which the post-decryption noise of approximate FHE is leveraged to provide DP guarantees [116], to the best of our knowledge, our work is the first one in which relaxing the accuracy of a homomorphic calculation *at the algorithmic level* is constructively usable as a degree of freedom to achieve better FHE performances and privacy guarantees.

The paper is organised as follows. First of all, we explore the related work in Section 2.2 and remind some preliminaries about HE and DP in Section 2.3. Then, we introduce and describe our argmax operator SHIELD in Section 2.4 and more specifically its FHE implementation in Section 2.5, before presenting SPEED application case in Section 2.6. Section 2.7 develops an analysis of SHIELD from the points of view of DP and HE. Finally, our experimental results are presented in Section 2.8.

2.2 . Related work

In [180], the authors survey recent works in which DP and cryptographic primitives take advantage of each other, either

- cryptography for DP: cryptographic primitives allow to get the privacy-utility trade-off of a standard DP mechanism but without the need of a trusted server [6, 49, 68, 80]. This is an improvement compared to *local DP* which, by making the data owners noise their data before outsourcing them, does not need a trusted server either but gives a poorer privacy-utility trade-off [108, 176]

- or DP for cryptography: design "leaky" cryptographic primitives that ensure DP and are more efficient than traditional primitives [16, 178, 179]

[16, 178, 179] are tailored to specific applications, respectively SQL queries, anonymous communication systems and oblivious RAM. Our work follows this line of DP for cryptography but in the context of election.

In [196], the authors propose an algorithm with a close goal, namely heavy-hitters (most frequent items) detection, which is inherently differentially private thanks to random sampling. Nevertheless, the goal of this inherent probabilistic behaviour is not computational efficiency since the method is not articulated with cryptographic primitives. Moreover, this algorithm works on sequential data. Even if it does not restrict its generality since any data can be seen as sequential, the utility does depend on the sequential representation of the data, which may not be optimal if there is no semantic value to this representation. Finally, the algorithm is iterative and thus requires a lot of communication with the users.

As far as federated learning is concerned, the work from [171] is interesting because it leverages the error induced by encryption to derive DP guarantees. The aggregation protocol is based on the security of LWE problem and on the Multi-Party Computation protocol of Packed Shamir secret sharing scheme [72]. Nevertheless, LWE is not used to directly encrypt the values of interest but rather to generate one-time pads of the same dimension of these values while only needing to communicate much smaller vectors to the server. These one-time pads allow a secure aggregation and DP guarantees are ensured by the error induced by LWE encryption.

2.3 . Preliminaries on BFV homomorphic cryptosystem

In this section we recall the general principles of the **BFV** homomorphic cryptosystem [69] which will be used in a batched manner. Since we know in advance the function to be evaluated homomorphically, we can stick to the somewhat homomorphic version described below. Let $R = \mathbb{Z}[x]/\Phi_m(x)$ denote the polynomial ring modulo the m-cyclotomic polynomial with $n = \varphi(m)$. The ciphertexts in the scheme are elements of polynomial ring R_q, where R_q is the set of polynomials in R with coefficients in $\mathbb{Z}_q$. The plaintexts are polynomials belonging to the ring $R_t = R/tR$. For $a \in R$, we denote by $[a]_q$ the element in R obtained by applying modulo q to all its coefficients. As such, **BFV** scheme is defined by the following probabilistic polynomial-time algorithms:

BFV.ParamGen(λ) $\rightarrow (n, q, t, \chi_{key}, \chi_{err}, w)$. It uses the security parameter λ to fix several other parameters such as n, the degree of the polynomials, the ciphertext modulus q, the plaintext modulus t, the error distributions, etc.

BFV.KeyGen$(n, q, t, \chi_{key}, \chi_{err}, w) \rightarrow (pk, sk, evk)$. Taking as input the parameters generated in **BFV.ParamGen**, it calculates the private, public and evaluation key. Besides the public and the private keys, an evaluation key is generated to be used during computation on ciphertexts in order to reduce the noise.

BFV.Enc$_{pk}(m) \rightarrow c = (c_0, c_1)$. For $m \in R_t$, compute the ciphertext $c = (c_0, c_1) \in R_q^2$, using the public key pk.

BFV.Dec$_{sk}(c) \rightarrow m$. It computes the plaintext m from the ciphertext c, using private key sk.

BFV.Add$(c_1, c_2) \rightarrow c_{add}$ with $c_{add} = \left([c_{1,0} + c_{2,0}]_q, [c_{1,1} + c_{2,1}]_q\right)$.

BFV.Mul$_{evk}(c_1, c_2) \rightarrow c_{mul} = (c_0, c_1, c_2)$ with $c_0 = \left[\left\lfloor \frac{t}{q} \cdot c_{1,0} \cdot c_{2,0} \right\rceil\right]_q$, $c_1 = \left[\left\lfloor \frac{t}{q} \cdot (c_{1,0} \cdot c_{2,1} + c_{1,1} \cdot c_{2,0}) \right\rceil\right]_q$ and $c_2 = \left[\left\lfloor \frac{t}{q} \cdot c_{1,1} \cdot c_{2,1} \right\rceil\right]_q$.

In order to reduce the number of elements in the ciphertexts obtained after a multiplication, a relinearisation method is proposed: **BFV.Rel**$(c_0, c_1, c_2) \rightarrow ct' = (c'_0, c'_1)$ such that $\left[c_0 + c_1 * sk + c_2 * s^2\right]_q = \left[c'_0 + c'_1 * sk + r\right]_q$ with the norm $||r||$ small.

For further details on the precise two relinearisation methods and the full description of the scheme, we refer the reader to the original paper [69]. Let us also note that to this original scheme, one can apply batching (also known as packing), an optimization method for FHE allowing to put several clear messages into a single ciphertext and execute parallel operations on them into a SIMD (Single Instructions Multiple Data) manner. The technique of ciphertext-packing is based on polynomial CRT (Chinese Reminder Theorem) and was originally described in [169], [32].

2.4 . SHIELD: Secure and Homomorphic Imperfect Election via Lightweight Design

In this paper, for any $m \in \mathbb{N}$, $[m]$ will denote the set $\{1, ..., m\}$ (which is, by convention, the empty set if $m = 0$).

Let K be the number of classes of the classification problem. Let n be the number of voters or teachers and, given a sample and $k \in [K]$, let n_k be the number of teachers who voted for class k.

2.4.1 . Principle of SHIELD

We propose a novel operator that can be viewed as an aggregation operator for categorical data, as well as a voting rule, or even a probabilistic argmax. This operator, called SHIELD (Secure and Homomorphic Imperfect Election via Lightweight Design) aims at computing the aggregation of categorical data - or equivalently the winner of an election - on a server while ensuring the privacy of the inputs from both the server and the end-users that may try to retrieve sensitive information from the output. Let us now formally introduce SHIELD.

First of all, SHIELD is meant to be computed in the homomorphic domain. Here are some notations we will use to describe its homomorphic behaviour. Enc and Dec respectively denote the encryption and decryption functions of some homomorphic encryption system defined on $\mathbb{Z}_2$. $\oplus$ and $\otimes$ respectively represent the homomorphic addition and multiplication. When these operators are applied on vectors, they denote the element-wise corresponding operations. Note that the negation of $x \in \mathbb{Z}_2$ is homomorphically performed via $\text{Enc}(1) \oplus \text{Enc}(x)$ and the homomorphic *or* operator, denoted $\curlyvee$, between $x \in \mathbb{Z}_2$ and $y \in \mathbb{Z}_2$ is performed via $[\text{Enc}(x) \oplus \text{Enc}(y)] \oplus [\text{Enc}(x) \otimes \text{Enc}(y)]$ and will be written $\text{Enc}(x) \curlyvee \text{Enc}(y)$ in the following.

Definition 7. *Let $K \in \mathbb{N}^*$. A vector $z \in (\mathbb{Z}_2)^K$ is said to be a* one-hot encoding *vector if there exists $k_0 \in [K]$ such that $z_{k_0} = 1$ and, for all $k \in [K] \setminus \{k_0\}$, $z_k = 0$. In this case, we say that z* codes *for the class k_0 or that z is the one-hot encoding of the class k_0.*

Let $(p, a) \in (\mathbb{N}^*)^2$. Let $\mathcal{S}_{p,a}$ denote SHIELD operator with parameters p and a, that we define in the following.

Let $(n, K) \in (\mathbb{N}^*)^2$ that we consider fixed in the remainder of this section. Let $Z = (\text{Enc}(z^{(i)}))_{i \in [n]}$ be a list of n encrypted one-hot encoding K-dimensional vectors, some of

these vectors being possibly equal (it is necessarily the case for some vectors when $K < n$). Then $\mathcal{S}_{p,a}(Z)$ is an encryption of one of the $z^{(i)}$, and with high probability (see Section 2.8 for quantitative results) $\mathcal{S}_{p,a}(Z)$ is an encryption of the most frequent of the one-hot encoding vectors of Z. $\mathcal{S}_{p,a}$ is formally defined in Algorithm 3 where, for the sake of clarity, we do not explicitly write the encryption function (e.g. $res = z^{(i_0)}$ instead of $res = \mathsf{Enc}(z^{(i_0)})$). $\mathcal{S}_{p,a}$ draws p vectors of Z *with replacement* in a uniformly random manner and multiply them. The resulting vector π is an encryption of the one-hot encoding of the class k_0, $k_0 \in [K]$, if all the p drawn encrypted vectors code for the same class k_0. Otherwise, π is the null vector of $(\mathbb{Z}_2)^K$. If a non-null vector has already been found, the current π is ignored (since the bit $found\_not\_null$ has been set to 1). Of course, since the algorithm is computed in the encrypted domain, it has to run until the end of the *for* loop but everything works as if the algorithm repeated this operation until it gets a non-null vector and then ignored the remaining product vectors. This first non-null vector is the output of $\mathcal{S}_{p,a}$. If no non-null vector was produced after a iterations, a null vector is output and we say that $\mathcal{S}_{p,a}$ *failed*.

Algorithm 3: SHIELD

Input : number of vectors n, number of classes K, list of encrypted votes Z, number of multiplications p, number of terms a
Output: $res = z^{(i_0)}$ where $i_0 \in [n]$

$res \leftarrow (0, \dots, 0) \in (\mathbb{Z}_2)^K$;
$found\_not\_null \leftarrow 0$;
for j *in* $[a]$ **do**
  $\pi \leftarrow (1, \dots, 1) \in (\mathbb{Z}_2)^K$;
  for l *in* $[p]$ **do**
    Draw a vector z of Z uniformly at random;
    $\pi \leftarrow \pi \otimes z$;
  end
  $res \leftarrow res \oplus (1 \oplus found\_not\_null) \otimes \pi$;
  $is\_not\_null \leftarrow \bigoplus_{k=1}^{K} \pi_k$;
  $found\_not\_null \leftarrow found\_not\_null \mathbin{\textcircled{\scriptsize \vee}} is\_not\_null$;
end

a being fixed, the choice of p must consider the tradeoff between, on one hand, the accuracy of the operator, e.g. the probability of getting the truly most frequent vector (see the considered accuracy metrics in Section 2.7.1), and, on the other hand, the probability of avoiding a failure and the computational complexity. Indeed, when p increases, the probability of getting a null vector (and then failing) increases, as well as the computational complexity, but the probability of getting the most frequent vector, knowing that the algorithm did not fail, increases too.

2.4.2 . Multi-degree SHIELD

We can imagine a parameter p that decreases as the iterations run, as if it adapted to the vote distribution. Indeed, on one hand, a high p for the first iterations ensures (with high probability) that we get the truly most frequent vector if getting a non-null vector is easy (i.e. probable), which happens if a vast majority of the vectors code for the same class (e.g. a vast

majority of voters agree on one candidate). On the other hand, if the first iterations failed, which suggests that getting a non-null vector is not so probable, the number p of multiplications decreases in order to make the production of a non-null vector easier. In this framework, our SHIELD operator can be represented by a polynomial $\sum_{p=1}^{D} a_p X^p$ with positive integer coefficients, where $\sum_{p=1}^{D} a_p = a$ and some a_p's may be null. We call $\sum_{p=1}^{D} a_p X^p \in \mathbb{N}[X]$ the *polynomial parameterisation* of SHIELD. There is indeed a bijection between the set of operators and $\mathbb{N}[X]$ since the order of the terms of different degrees is constrained to be the one of decreasing degrees. Nevertheless, the analogy seems to stop here since the algebraic structure of $\mathbb{N}[X]$ does not apply to the set of operators (think about a factorisation like $X^2 \sum_{p=0}^{D} a_p X^{p-2}$, that would draw for once two vectors and use them for all the a terms, whereas we here want to independently draw the vectors for each term).

Note that we can easily ensure that multi-degree SHIELD does not fail by imposing $a_1 = 1$. Indeed, when we draw only one one-hot encoding vector, without multiplying it with others, we cannot get a null vector. Moreover, $a_1 > 1$ is useless since the first draw of a single vector will succeed.

It is easily seen that multi-degree SHIELD is a generalization of SHIELD and, as such, in the remainder of this article, multi-degree SHIELD will simply be referred to as SHIELD.

2.4.3 . Offset parameter

The SHIELD operator as defined above cannot always provide finite DP guarantees. Let us consider two adjacent databases d and d' such that, in d, a class c was chosen by no voter and, in d', c was chosen by one voter. Then, with input d, SHIELD will never output c because it cannot pick a one-hot encoding for c, the probability of outputting c is then null. On the contrary, with input d', there is a non-null probability (even if it is small) of outputting c. Hence, the ratio of probabilities of outputting c is not bounded and we get an infinite privacy cost.

To avoid this problem, we force all the classes to have at least one vote by creating a dummy one-hot encoding for each class. More generally, ω dummy one-hot encodings can be created for each class, where ω is another parameter of SHIELD, called the *offset*.

Algorithm 4 gives the pseudocode of the multi-degree version of SHIELD with the offset parameter.

In our experiments, we fixed ω to 1, letting the optimisation of this parameter for further work. It is nevertheless intuitive that the greater ω, the worse the accuracy because, when ω is large, the distribution of the votes is flattened and the probability of outputting the true argmax is lower.

2.4.4 . Exponential argmax operator

As an inherently stochastic mechanism that does not resort to noise addition but rather outputs a value with a probability that is an increasing function of its utility (if we deem that the vote frequency of a class constitutes its utility), SHIELD can be compared to the exponential mechanism (introduced in [129]) which samples its output following the softmax distribution of the utility. However, the sampling in the encrypted domain constrains the shape of the probability distribution and introduces a dependency of the practically implementable distributions with the computational efficiency of the operator.

Algorithm 4: Multi-degree SHIELD

Input : number of vectors n, number of classes K, list of encrypted votes Z, polynomial $(a_p)_{p\in[D]}$, offset ω
Output: $res = z^{(i_0)}$ where $i_0 \in [n]$

$Z \leftarrow Z$ augmented by ω encrypted one-hot encodings for each class;
$res \leftarrow (0,\dots,0) \in (\mathbb{Z}_2)^K$;
$found\_not\_null \leftarrow 0$;
for p *in* $[D]$ **do**
  for j *in* $[a_p]$ **do**
    $\pi \leftarrow (1,\dots,1) \in (\mathbb{Z}_2)^K$;
    for l *in* $[p]$ **do**
      Draw a vector z of Z uniformly at random;
      $\pi \leftarrow \pi \otimes z$;
    end
    $res \leftarrow res \oplus (1 \oplus found\_not\_null) \otimes \pi$;
    $is\_not\_null \leftarrow \bigoplus_{k=1}^{K} \pi_k$;
    $found\_not\_null \leftarrow found\_not\_null \veebar is\_not\_null$;
  end
end

Note that softmax has been approximately implemented in FHE through polynomial approximation [115] but this requires a quite high multiplicative depth (with a polynomial of degree 12 for approximating the exponential function and even more for approximating the inverse function) and results in a significant computational overhead. Moreover, using such an implementation would still require additional homomorphic operations like comparisons to actually sample the output according to this distribution.

Rather, a sampling method that follows the exponential distribution by construction, in the spirit of SHIELD as presented in this paper, would be more seducing. Sampling each vote independently with a fixed probability would actually yield an output distribution that exponentially depends on the vote frequencies but it seems that the probability of failing by not outputting any class would be quite high for practical parameters. We let further work on this question as a perspective.

2.5 . FHE implementation of SHIELD

Algorithm 4 is a generic version of SHIELD that actually needs to be adapted for an implementation using an HE cryptosystem. And first, there are two kinds of possible encodings depending on the encryption scheme that is used:

- Single Instruction, Multiple Data (SIMD). Using the BFV cryptosystem, a number of values are encoded simultaneously in a polynomial which is then encrypted. A single operation on a ciphertext leads to the same operation applied to all values encoded inside the ciphertext.
- Single Instruction, Single Data (SISD). One way of using the TFHE cryptosystem is to

use a single ciphertext to encrypt a single value. This is less efficient than using SIMD but unlocks a set of complex operations on that ciphertext that are impossible to implement otherwise.

We implement SHIELD with two separate methods: one uses the BFV cryptosystem with SIMD operations; the other uses the TFHE cryptosystem with SISD operations.

2.5.1 . Implementing SIMD-SHIELD

Although using BFV allows us to speed up SHIELD considerably by batching different samples together in the same ciphertext, some constraints require adapting parts of Algorithm 4 for them to work.

a. Multiplicative depth. As it is the case for other similar HE schemes, we need to set the parameters of BFV according to the multiplicative depth of the computation. The higher the multiplicative depth, the larger the parameters, and the less efficient the overall computation. For this reason, some parts of the algorithm, like Line 9, need to be changed. We can store all of the values for Enc(z) over the loop and multiply them in a classic tree-based approach (instead of multiplying them sequentially) which reduces the multiplicative depth of the computation from p to $\log_2(p)$.

The same change is applied everywhere it is needed, that is to say at Lines 11 and 13 of Algorithm 4.

b. Selecting the teacher. Selecting the voter, also called teacher because of SPEED application case (see Section 2.6), at lines 8 and 9 of Algorithm 4 is easy enough when the SHIELD algorithm is called for a single sample at once. However, in order to speed up the algorithm and make use of the SIMD property of the BFV cryptosystem fully, we actually run the SHIELD algorithm for a number of samples at a time.

For instance, if $\pi^{(i)}$ is the π vector of K values for sample i, then the actual vector encoded in the ciphertext for the packed algorithm would be

$$\pi = \left(\pi_1^{(1)}, \ldots, \pi_K^{(1)}, \pi_1^{(2)}, \ldots, \pi_K^{(2)}, \ldots\right) \tag{2.1}$$

This allows us to use the full size of the polynomials we encrypt. These polynomials have degrees in the order of $\approx 2^{15}$ while K is usually in the order of ≈ 10.

Therefore the teacher selection step has to be modified. The new encoding of teachers t's vote for sample i is:

$$\left(0, \ldots, 0|0, \ldots, 0| \ldots |z_t^{(i)}| \ldots |0, \ldots, 0|\right) \in \mathbb{B}^{N \times K}$$

which is a vector with N slots of K binary values where $z_t^{(i)}$ is teacher t's original one-hot encoded vote for sample i. It is located at the i^{th} slot of the encoding. From now on we'll call $z_t^{(i)}$ this new encoding of the teacher's votes. Algorithm 5 presents the process for teacher selection and creation of the π vector using this new encoding.

Algorithm 5: Teacher selection. With n the total number of teachers, this algorithm describes the actual steps for selecting the teachers that get to vote in the SIMD encoding paradigm.

```
1  for j in [a_p] do
2      for l in [p] do
3          for t in [n] do
4              z_t ← (0, ..., 0);
5          end
6          for i in [N] do
7              Draw a vector z_t^(i) of Z uniformly at random;
8              z_t ← z_t ⊕ z_t^(i) ;
9              update m_t ;
10         end
11         π ← (1, ..., 1);
12         for t in [n] do
13             z_t ← z_t ⊕ m_t ;
14             π ← π ⊗ z_t ;
15         end
16     end
17 end
```

At Line 9 a mask m_t is updated. For every teacher t, the mask m_t is a plaintext vector that contains 0s in the place of samples for which the teacher is selected and 1s in the place of samples for which the teacher is not selected. As an example, for $K = 2$ and $N = 4$, if teacher t votes for samples 1 and 3, then $m_t = (0, 0|1, 1|0, 0|1, 1)$.

This mask is then added to z_t before the multiplication to the π vector so that all the samples for which the teacher is not selected do not impact the result: their slots are filled by ones. If the mask is not used, then all non-selected slots will be filled with 0s and therefore would set everything to 0 after the multiplication.

For this multiplication, as mentioned before, we opt to store all of the z_t vectors and create a multiplication tree to reduce the multiplicative depth.

c. Rotations. One other constraint that schemes such as BFV suffer from, is that it is very hard and costly to extract certain values from the ciphertext to apply an operation *only* to them. Such is the case when trying to implement Line 12 in Algorithm 4. The individual π_k values cannot be extracted and summed together in a straight-forward manner. One thing we can do however, at a relatively low cost (both in terms of performance and noise inside the ciphertext), is to rotate the vector encoded in the ciphertext. This leads to an implementation of Line 12 that we present using the example $\pi^{(i)} = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$.

$$
\begin{aligned}
&(0,0,0,1,0,0,0,0,0,0)\\
+&(0,0,0,0,0,?,?,?,?,?) && \leftarrow \text{rotate by 5}\\
=&(0,0,0,1,0,?,?,?,?,?)\\
+&(0,1,0,?,?,?,?,?,?,?) && \leftarrow \text{rotate by 2}\\
=&\dots
\end{aligned}
$$

One can see how, using $\log_2(K)$ rotations and sums, we can obtain $\sum_j \pi_j^{(i)}$ in the *first* coordinate of the $\pi^{(i)}$ vector. The question marks ? represent values that are rotated over from the next slot, (recall the complete form of π in equation 2.1).

Therefore, we cannot control the values in the rest of the coordinates. And this is not enough. For Line 13 to work, we need to have a vector where *all* coordinates $\pi_j^{(i)}$ are filled with $\sum_j \pi_j^{(i)}$, not just the first one. To obtain this, we have to multiply by a plaintext with values $(1,0,0,\dots)$ to select only for the first coordinate of π and then re-populate the rest of the coordinates using rotations and sums exactly in the opposite way as used for the computation of the sum of the $\pi_j^{(i)}$ values.

d. Packing the polynomial rounds together. Up until now, for clarity, we presented a version of our algorithm that packed all or some of the N samples together in a single ciphertext. In practice, to speed up the computation further, we also pack the polynomial rounds together. What we mean by "polynomial rounds" is the two *for* loops at Lines 1 and 2 in Algorithm 5. We can remove these *for* loops and compute them in parallel in a single ciphertext.

2.6 . An application case: SPEED

2.6.1 . SPEED workflow

Our SHIELD operator is actually tailored to a learning protocol called SPEED, from [83], itself inspired from PATE [141]. SPEED method is illustrated by Figure 2.1, inspired from [83]. Assuming the existence of a public unlabelled database Δ (we will keep this notation throughout the paper), SPEED enables several data-owners, called *teachers*, to collaboratively train a classification model without outsourcing their data that are considered private. The idea is to label Δ and use it to train the final classification model, called the *student* model or simply the student. To do so, each teacher is asked to train a model beforehand for the same task as the student's target task with its own data only and, for each sample of Δ to label, every teacher infers a label through their model and sends this label to an aggregation server. The server then counts the number of labels received for each class, also seen as votes, and outputs the dominant class which is sent to the student for training.

As it was described, the protocol does not protect the data from the server or the end-users. Before explaining how data privacy is ensured, let us present the threat model.

2.6.2 . Threat model

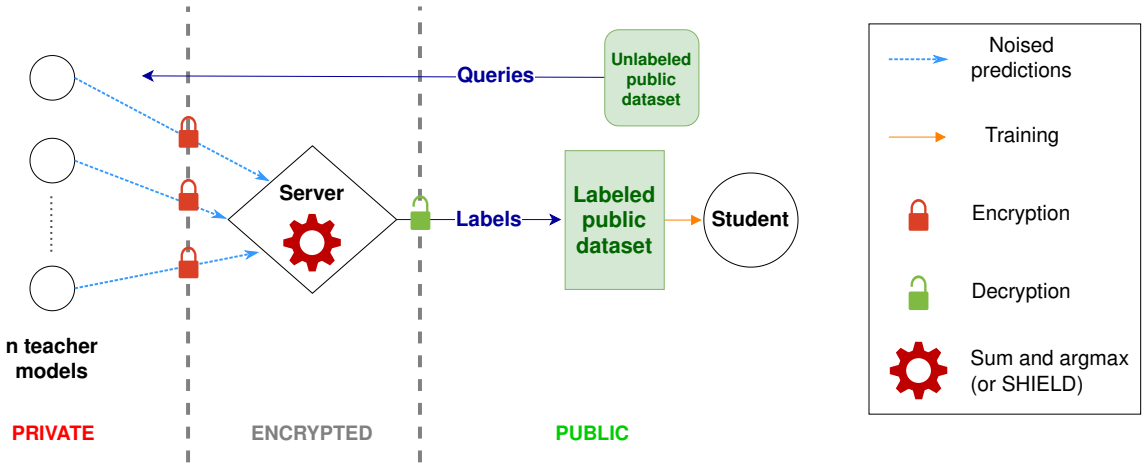

Figure 2.1: SPEED learning protocol

All the actors of the protocol, namely the teachers, the server and the student are considered honest-but-curious. This means that they execute their task correctly but may use the data they have access to to retrieve sensitive information about the teachers' data. The end-users are also considered curious, the honest part not being relevant for end-users that are not involved in the training. Note that, in many real-life cases, the teachers may be end-users of the student model.

A limitation to the threat model is that the server is not considered to have access to the trained student model since our DP analysis assumes that the adversary only sees the output class, which is not exactly the case of the server (see 2.7.2 for more details).

2.6.3 . Data protection

To prevent the student and *a fortiori* the end-users (by postprocessing) from discovering sensitive information by attacks such as e.g. model inversion or membership inference, we apply DP. The teachers noise their votes before sending them to the server.

One could argue that the noise added by the teachers would also blur the sensitive information to the server. Nevertheless, the added noise is precisely scaled so that it protects the output of the aggregation, i.e. the dominant class, without harming too much the student accuracy. If the individual votes sent to the server were to be protected by DP *before* aggregation, thus achieving what is called *local DP* [62, 107, 108], this would require much more noise, too much noise to ensure a reasonable accuracy for the student model. As a consequence, the votes need to be protected from the server another way. This is where homomorphic encryption makes its entrance. After noising their votes, the teachers encrypt them. The server then receives the encrypted votes and perform their aggregation (sum and argmax) in the homomorphic layer. Finally, the output of the aggregation is sent to the student that owns the decryption key and is therefore able to decrypt it.

By the honest-but-curious hypothesis, we then assume that:

- the teachers send their votes correctly noised and encrypted to the server

- the server performs the aggregation in the homomorphic domain as it is asked to
- the student decrypts the data and get trained; importantly, it does not share the decryption key with the server.

A real-life scenario could involve hospitals that own patients' medical data and aim at training a global model that would help the early diagnosis of a specific disease. In this case, the end-users would be the hospitals themselves.

2.6.4 . Faster SPEED with SHIELD

Our SHIELD operator can be used to replace the sum and argmax computations on the server side in SPEED (represented by the gear wheel in Figure 2.1). After receiving all the votes from the teachers, the server randomly picks some vectors with replacement as described in Section 2.4.1. Note that, being honest-but-curious, the server is trusted to compute SHIELD without mistake. Interestingly, the rest of SPEED protocol remains unchanged, except the sending of dummy one-hot encodings by some teachers, according to the offset parameter (see Section 2.4.3).

2.7 . Analysis of SHIELD

2.7.1 . *A priori* accuracy metrics

The ultimate accuracy that we want to maximize in SPEED application case is obviously the testing accuracy of the student model. Nevertheless, it could be interesting to measure the accuracy of the argmax operator itself, independently of the student training. Also, even if this depends on the teachers' votes and thus on the used dataset, this enables us to evaluate polynomial parameterizations without performing the student training, which is much faster and allows to test much more parameterizations. We call such an accuracy an *a priori* accuracy.

The most straightforward way to define the argmax accuracy is probably to consider the probability of getting the exact argmax. Nevertheless, this approach treats any mistake the same way. It could be argued that outputting, say, the class that received the second greatest number of votes is better than outputting the least preferred class. Taking such a concern into account in our metric would also give a better hint about the student accuracy since, while the most preferred class (i.e. the exact argmax) is not always the ground truth class, a class with a lot of votes is more likely to be the ground truth class.

We could then make the assumption that the frequency of votes for a class is proportional to the probability of this class being the ground truth class of the sample (which is not necessarily the most preferred class). This would correspond to an assumption of well-calibrated vote distributions. In this context, another accuracy metric would be the probability of outputting the ground truth class of the sample. We call this metric the *ground truth accuracy*, since it does not focus on outputting the exact argmax but rather the ground truth class. If p_k denotes the probability of SHIELD outputting class k, for $k \in [K]$, the ground truth accuracy, written

GTA, is:

$$\text{GTA} = \sum_{k=1}^{K} \frac{n_k}{n} p_k.$$

Of course, both metrics must be averaged on all the samples sent to the teachers.

2.7.2 . Differential privacy analysis

Since the student model training requires many requests to the teachers and, indirectly, to their private datasets, we use, as in [83], the moments accountant technique [2] to get a better privacy cost over composition.

We here consider that two databases d and d' are adjacent if they are the concatenations of the datasets from the same number of teachers and only one teacher differs from one database to the other. This implies that either all the n'_k, counts for database d', for $k \in [M]$, are equal to the n_k, counts for database d, in which case the corresponding moments accountant is null, or the n'_k differ from the n_k only for two values of k, say k_1 and k_2, such that $n'_{k_1} = n_{k_1} - 1$ and $n'_{k_2} = n_{k_2} + 1$ (i.e. the differing teacher votes for k_1 in d and k_2 in d').

The stochastic behaviour of our operator uncommonly does not come from an additional random noise, since the operator is inherently probabilistic. This is this very property of our operator that we leverage to ensure DP. Computing the privacy cost of the training, as well as the *a priori* accuracy, thus requires knowing the probabilities of outputting each class.

2.7.3 . Computing the probability distribution of the output

We compute the probability distribution of the output of the algorithm SHIELD with a given polynomial parameterisation in a recursive manner.

For a sample x of Δ, let $\mathcal{A}_{P,x}$ be the mechanism that takes the whole database (concatenation of the teachers' datasets) as input and outputs the class sent to the student i.e. the output of SHIELD, with the polynomial parameterization $P \in \mathbb{N}[X]$.

Let d be the database composed of the teachers' data. Let k be a class of the problem.

If $P = X$, $\mathbb{P}[\mathcal{A}_{P,x}(d) = k] = \frac{n_k}{n}$.

If $P = X^p + Q(X)$, where $Q \in \mathbb{N}[\mathbb{X}]$ and $p \in \mathbb{N}^*$ is greater or equal than the degree of Q,

$$\mathbb{P}[\mathcal{A}_{P,x}(d) = k] = \left(\frac{n_k}{n}\right)^p + \left(1 - \sum_{j=1}^{K} \left(\frac{n_j}{n}\right)^p\right) \mathbb{P}[\mathcal{A}_{Q,x}(d) = k].$$

Using these expressions, we simply compute the moments accountant for each query by taking the maximum over all pairs (d, d') such that d is the database constituted by the concatenation of the teachers' database and d' is a database adjacent to d. We then derive the overall privacy cost using Theorems 1 and 2.

Note that the obtained DP guarantees are *data-dependent* since we explored only the pairs of adjacent databases such that one of them is the actual database given by our application. The very values ϵ and δ of these guarantees then reveal some information about the training data. In a real-life scenario, these values should be sanitized before being published, as in [142] for instance, but this is beyond the scope of this work.

2.7.4 . The differential privacy analysis does not apply to the server

When we compute the probabilities of outputting a class, we do not suppose anything about whose votes are drawn i.e. we do not condition the probabilities on some particular drawing event. This amounts to assume that the adversary only sees the output class, and does not know, in particular, which teachers were selected in the sampling. This assumption cannot apply to the server since it draws the one-hot encodings itself and knows which teacher they come from, for having receiving the encodings one by one from the teachers.

To give an insight of why this subtlety is problematic, let us propose some concrete situations where the DP guarantees are obviously not protecting the vote of the server's victim, i.e. the teacher whose vote the server wants to know.

- With the polynomial parameterization $X^k + X$, $k \in \mathbb{N} * \setminus \{1\}$, if the server draws $k-1$ teachers and its victim for the term X^k and then its victim for the term X, then the server will know that the class sent to the student is its victim's vote.

- Supposing that the server knows the votes of all the teachers except its victim's (classical assumption in DP), it will be able to recover its victim's votes in many cases. For instance, with the polynomial parameterization X^k+X, $k \in \mathbb{N}^* \setminus \{1\}$, if the server draws k teachers who do not all have the same vote for the term X^k and its victim for the term X, then the class sent to the student is its victim's vote.

To address this vulnerability, we could think of an additional entity that receives the votes from the teachers and shuffles them before sending them to the server. However, the server would know if a same vote was drawn several times (remind that the drawing is with replacement), which still constitutes some information we did not account for in our DP analysis. Suppose that the server knows that all the teachers except its victim voted for a class c. Moreover, suppose that the offset parameter is set to 1 and that there are $|C|$ classes in the problem. Then, there are $|C|-1$ votes different than c and the victim's vote, which is unknown. Assume that the polynomial parameterization is $|C|X^2 + X$. If the $2|C|+1$ votes that the server drew are all from different sources - teachers or dummy one-hot encodings - (remind that the server knows it) and the output class is not c, then the server knows with certainty that its victim did not vote for c (otherwise, there would have been $|C|+1$ drawn votes for c and, among the $|C|$ pairs the server drew for the term in X^2, no pair would have been composed of two identical votes different from c and at least one pair would have been composed of two votes for c and then the output would have been c).

These observations show that we need to constrain the server not to see the student model once it is trained. Note that the information leakage induced by the server's knowledge may not jeopardize much the data privacy in practice. We only argue here that our DP analysis does not allow us to derive DP guarantees from the point of view of the server, which might be possible with a more involved (and likely quite complex) analysis, although with probably worse guarantees.

2.7.5 . Extension of the threat model

We could extend the threat model and assume that the server has access to the final model by designing a more complex algorithm for which the teachers would be homomorphically selected via encrypted masks.

Another interesting idea mentioned above would be to make use of an intermediate entity that would shuffle the encrypted votes before the server receives them, with inspiration from the ESA (Encode, Shuffle, Analyze) method from [22]. Nevertheless, the server would still know if it selected several times the same teacher, even without knowing which one it is, and this is still theoretically an information leakage that is not simple to analyze (cf. Section 2.7.4). A way to solve this issue and to actually leverage the anonymity provided by the shuffling would be to design an algorithm that uses sampling *without* replacement and to force the teachers to send a new encryption of their votes for each polynomial rounds, which would significantly increase the complexity of the protocol and its communication cost.

Aware of this weakness of our threat model compared to SPEED's one in [83], we let these improvements for further work.

2.7.6 . Computational complexity of SHIELD

Compared with previous argmax HE computation methods, SHIELD is unique in that its complexity only linearly depends on the number of classes for the chosen machine learning problem. Indeed, the main impact of an increase in the number of classes is that the encoding space increases by the same amount (and therefore the time overhead is linear). A secondary impact is the logarithmic increase in the number of rotations needed for the computation of $\sum_j \pi_j^{(i)}$ as seen in Section 2.5.1. All previous work uses one (or a combination) of two methods to evaluate an exact argmax over a number of values: a tournament method or a league method. We refer the reader to [43, 97] for specific implementation details. Here we focus on their complexity with respect to the number of classes.

- a league is a system of comparison where every value is compared with every other value. The winner is the value that was greater than every other one. Think of a football league like French first division league ("Ligue 1") for this kind of system. The use of a league method yields a quadratic complexity in the number of classes. This leads to very high performance overheads as the number of classes increases. However, contrary to the tournament method, increasing the number of classes does not affect the multiplicative depth of the circuit to be evaluated. This is what makes this method useful in the homomorphic domain in spite of its complexity.

- a tournament is a system where values are compared two-by-two and the losers are discarded at every round. Think of the FIFA World Cup for this kind of system. Using a tournament method has a - theoretical - linear complexity in the number of classes. In practice, this is not the case. As the number of classes increases, the comparison tree used for the evaluation increases in depth logarithmically. For levelled homomorphic schemes such as BFV or BGV (those we use in this article) used in [97], this means an increase in parameter size to match the multiplicative depth of the new tree. In turn, this impacts the performance of the overall scheme on top of the theoretical linear increase.

After a given point, the increase in parameter size becomes prohibitive and one needs to resort to finishing the computation using a league method as they do in [97].

Compared to all other existing works therefore, ours scales much better with the number of classes and therefore fits particularly well with use-cases with high numbers of classes.

2.8 . Experimental results

2.8.1 . Choice of the polynomial parameterization

We tested SPEED with SHIELD on MNIST dataset [114]. While the offset parameter has been set to 1, a key aspect of our experiments is the choice of a polynomial parameterisation that realises a good trade-off between model accuracy, DP guarantees and computational efficiency. Since the computational time overall depends on the sum of coefficients and the degree of the polynomial parameterisation, we proceeded by constraining the maximum degree and the maximum value for the sum of coefficients of the polynomials. We fixed the maximum degree to 4 because higher degrees resulted in too high computational complexity. For several integer values (6, 12, 17, 32), we considered all the polynomials of degree at most 4 whose sum of coefficients is less than this value. We do not go beyond a sum of coefficients equal to 32 to keep the computational time low. We then computed the DP guarantee ϵ, $\delta = 10^{-5}$ being fixed, for each polynomial, as well as its GTA that acts as a proxy for the student model accuracy which could not be determined in reasonable time for so many polynomials. Finally, we focused on the polynomials belonging to the Pareto front for these two criteria - DP guarantee ϵ and GTA - and picked the ones that yielded among the best DP guarantees without harming the accuracy too much. In practice, as it can be seen on Figure 2.2 the DP guarantee guided more our choice because the GTA, besides being only a heuristic for the actual student model accuracy, did not vary much among the polynomials of the Pareto front. Note that the GTA of the exact argmax is 72.35%. The chosen polynomials are respectively $2X^3 + 3X^2 + X$, $2X^4 + 6X^3 + 3X^2 + X$, $6X^4 + 6X^3 + 4X^2 + X$ and $8X^4 + 6X^3 + 4X^2 + X$ for a sum of coefficients of at most 6, 12, 17, 32[1]. We did not display the Pareto front for a sum of coefficients of at most 6 because it only contains one polynomial.

Table 2.1 displays the GTA, the student model accuracy and the DP guarantee ϵ for the chosen polynomial parameterizations, $\delta = 10^{-5}$ being fixed. The GTA and DP guarantee are averaged on the whole set of 8000 samples used for semi-supervised training, the DP guarantee being remultiplied by 100, the number of actual queries to the teachers. The student model accuracy is averaged over ten runs, each of which used a different random subset of 100 samples as labelled samples. The table also displays the number of correctly labelled samples (comparing to the ground truth label) out of the 8000 samples. The variance of the model accuracy among the runs is quite important and may explain why the accuracy surprisingly does not increase when the polynomial is better in terms of both GTA and number of correct labels.

[1]The chosen polynomial among the ones with a sum of coefficients at most 32 has a sum of coefficients equal to 19 only. This is good news for computational complexity because it allows us to batch all samples into a single ciphertext and therefore optimise the computation.

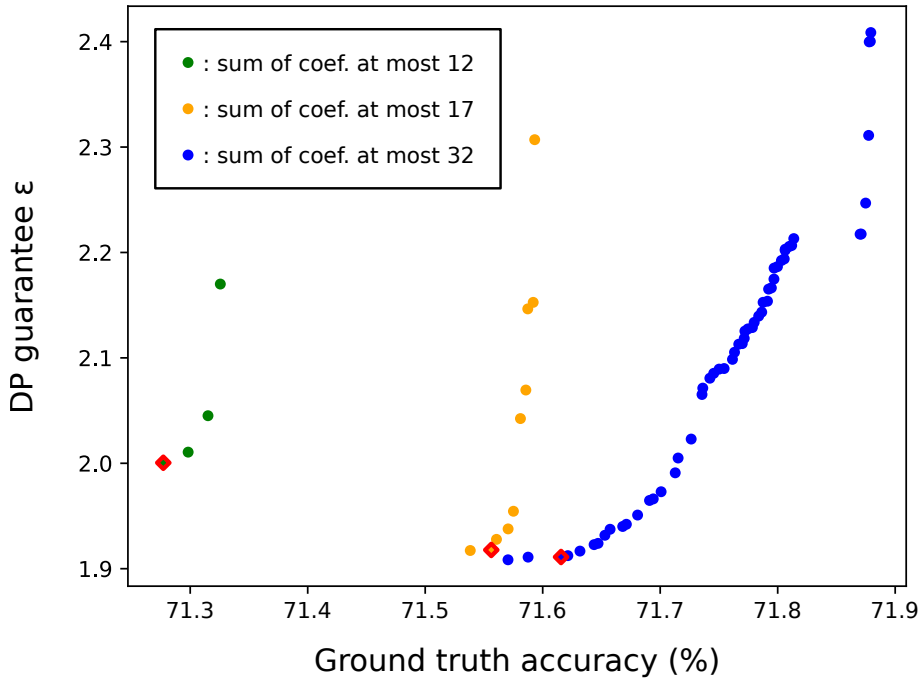

Figure 2.2: Pareto fronts of the polynomials for a fixed maximum sum of coefficients. The polynomials we chose for running the student model training are indicated by red-edged diamonds.

polynomial	GTA	number of correct labels	model accuracy	ϵ
exact argmax	72.35%	7516 (93.95%)	95.36%	∞
$2X^3 + 3X^2 + X$	70.06%	7166 (89.58%)	90.91%	2.39
$2X^4 + 6X^3 + 3X^2 + X$	71.26%	7327 (91.59%)	94.66%	2
$6X^4 + 6X^3 + 4X^2 + X$	71.56%	7358 (91.98%)	93.39%	1.92
$8X^4 + 6X^3 + 4X^2 + X$	71.62%	7367 (92.09%)	93.15%	1.91

Table 2.1: Accuracy and DP guarantee (with $\delta = 10^{-5}$) obtained with several polynomial parameterizations.

polynomial	samples	time (s)	time/sample (s)
$2X^3 + 3X^2 + X$	100	87.2	0.87
	341	112	0.33
$2X^4 + 6X^3 + 3X^2 + X$	100	123	1.23
	143	135	0.94
$6X^4 + 6X^3 + 4X^2 + X$	100	138	1.38
$8X^4 + 6X^3 + 4X^2 + X$	100	144	1.44
paper	samples	time (s)	time/sample (s)
[83]	100	390	3.9
[83] + [43]	100	160	1.6
[97]*	100	152	1.52
	5220	152	0.03

Table 2.2: Performance for the SIMD implementation of SHIELD (for 10 classes) for different polynomial parameterizations compared with previous work implementing exact argmax computations.
$*$ Times for [97] are presented but cannot directly compare with our results for reasons that are expanded upon below.

2.8.2 . SIMD SHIELD with BFV

For our implementation of the SIMD SHIELD algorithm, we use the BFV cryptosystem in the openFHE library [12]. The parameters we choose are the following: $\log_2(q) = 540$; $p = 65537$; $m = 65536$; $N = 32768$. These parameters achieve a security level of $\lambda = 128$ bits with a standard deviation of 3.2. Our implementation was tested on a machine with an AMD Opteron(tm) Processor 6172 using a *single thread*.

We achieve performances presented in Table 2.2 for a set of different polynomial parameterisations. Although we tested using the MNIST data set, the performance of an HE algorithm does not depend on the underlying data *by construction*. Otherwise one could infer something on the data from seeing the computation happen in the encrypted domain. For our implementation, we need to run the SHIELD algorithm over 100 samples. In the table however, we also present computation times for the case whereby we optimise the batching space with a higher number of samples to give an idea of what computation times could be achieved by optimising parameters further. For now, these optimisations are not yet possible in keeping with the Homomorphic Encryption Security Standard [7] which recommends the use of power-of-two cyclotomic polynomials. A new standard is reported to be in the works which would open applications to the secure use of non-power-of-two cyclotomic polynomials. That would allow us to optimise our parameters further.

Table 2.2 also compares our method with previous existing methods for *exact argmax computations*. Among these methods, the one presented in [83] as well as its later improvement in [43] perform worse overall for all polynomial parameterizations that we tested. It is important to note that these methods do not use batching by construction. Therefore the time per sample is fixed and does not depend on the amount of samples processed.

polynomial	samples	time (s)	time/sample (s)
$2X^3 + 3X^2 + X$	100	495,3	4,95
$2X^4 + 6X^3 + 3X^2 + X$	100	14287,8	14,29
$6X^4 + 6X^3 + 4X^2 + X$	100	20936,7	20,93

Table 2.3: Performance for the Cingulata with TFHE implementation of SHIELD

[97] on the other hand does make use of batching. In effect, by construction, they are constrained to batching sizes much higher than ours, therefore an amortised time of 0.03s could not be obtained over 100 samples. Times in Table 2.2 for [97] are taken from their Table 4 because it most closely matches our use-case. However important differences remain: we report their timings for 8 classes as it is the closest to 10 in the Table; timings are for a minimum computation, which is less time-consuming than an argmin computation, but no times are given for an argmin in the paper.

2.8.3 . Bitwise (SISD) SHIELD with Cingulata

To show the interest of the batching approach, we also implemented the basic version of SHIELD, as described in Alg. 4, with Cingulata crypto-compiler and its TFHE back-end.

Let us remind that Cingulata, formerly known as Armadillo [39], is a toolchain and runtime environment (RTE) for implementing applications running over homomorphic encryption. Cingulata provides high-level abstractions and tools to facilitate the implementation and the execution of privacy-preserving applications expressed as Boolean circuits.

Table 2.3 shows the execution times of SHIELD for different polynomial parameterisations when performed in a SISD fashion with TFHE and Cingulata. The experiments were performed with a single thread on an Intel Xeon processor with 16 GB of memory and Ubuntu 20.04 operating system. As shown in the table, the execution time of SHIELD increases with the degree of the polynomial and the sum of the polynomial coefficients. As expected, the overall performances are highly below the ones obtained when using BFV and its batching capabilities.

2.9 . Conclusion and perspectives

We proposed SHIELD, a homomorphic stochastic operator whose lightweight design necessary for fast homomorphic computations yields DP as a natural "by-product". This work reconciliates two complementary but usually independent - or even mutually constraining - privacy tools in an all-in-one operator whose inaccuracy is a crucial feature.

We hope this work will encourage new works on the design of private algorithms where FHE (or other cryptographic primitives) and DP leverage the advantages of each other. For instance, developing algorithms that would be useful in other settings than an election and broaden the scope of machine learning applications seems promising. In this perspective, an argmax algorithm that takes an histogram of the votes as input rather than the "physical" votes represented as vectors would have a more general applicability.

Testing SHIELD on more difficult datasets and especially datasets with numerous classes

could reveal its full potential. Besides, a more thorough theoretical study to get results that may lead us through the choice of the parameters (polynomial, offset) is desirable. Other versions including sampling without replacement (Section 2.7.5) or an exponential version of SHIELD (Section 2.4.4) would also deserve theoretical and experimental analyses. Studying SHIELD in terms of strategy-proofness and fairness could be interesting too and would extend the added value of SHIELD to the area of computational social choice and voting rules.

D - On counter-productive noise for data-dependent differential privacy guarantees

D.1 . Null data-dependent privacy cost of the exact argmax

While doing experiments on a subset of MNIST with polynomial parameterizations that yield better and better accuracies (up to the probability of getting the true argmax being more than 99,99%) we remarked that the value epsilon of the privacy cost did not increase much and did not seem to approach infinity. This surprising result suggested that the exact argmax operator had a finite privacy cost. Actually, on the subset we were working on, for every sample, the dominant class had at least two more votes than the second dominant class. We will say in the following that the distribution has a *highly dominant* argmax. This implies that, any database which is adjacent (i.e. differs from at most one teacher) to the database d we were working on has the same dominant class as d for every sample. As a consequence, the output of the argmax does not leak *any* information about which of two adjacent databases was used as input. In other words, the privacy cost of the exact argmax operator is null in this case.

D.2 . Counter-productivity of the noise regarding privacy

On the contrary, the so-called private argmax operator (noised by an additional random noise as in PATE [141, 142] and SPEED [83] or intrinsically stochastic as in SHIELD) may output any class and the probabilities of outputting a class depends on the frequencies of the votes for all classes. As a consequence, even changing only one teacher will change the probabilities of outputting some (or rather all) of the classes, even if the effect is mild. Therefore, the output of the DP argmax operator does give information on the probability of outputting a class and then on the frequencies of the classes in the votes. We end up in a (particular) situation where applying noise is counter-productive in the sense that it increases the privacy cost of revealing the output (by an infinite factor actually). Note, however, that this was not the case for the entire MNIST training set but only for a certain subset of it.

D.3 . The case of data-independent DP guarantees

This consideration only applies to *data-dependent* DP guarantees. In the data-independent case, the privacy cost of the exact argmax would be infinite because we would consider the maximum over all the pairs of adjacent databases i.e. all the possible pairs of distributions of n votes among K classes that differ by one vote on two classes. In this perspective, the question of the definition domain of the databases is crucial. Only giving data-dependent DP guarantees for the aforementioned subset of MNIST dataset, where, for every sample, the vote distribution has a highly dominant argmax, amounts to give a data-independent DP guarantee with a definition domain of the databases included in the set of databases such that the vote

distributions have a highly dominant argmax. This is obviously restricting the problem to a too easy subset of situations, and, as we showed above, this restricted problem is trivially solved by the deterministic exact argmax.

D.4 . Example of the age's sign

The noise addition degrading privacy guarantees is very counter-intuitive and may surprise *a priori*. Let us take a simple example to understand how the noise affects privacy. Revealing the sign of the age of a person is infinitely private (epsilon and delta null) if we assume that the adversary already knows that a person must have a positive age (quite natural assumption!). Imagine now that we noise the age with a unimodal noise, whose mode is zero, say a Gaussian noise, before computing the sign. The lesser the unnoised age, the more likely the sign of the noised age will be negative. This implies that revealing the sign of the noised age does leak some information about the unnoised age. Clearly, the noise addition does harm the privacy guarantees in this case. Nevertheless, note that this does not contradict the post-processing immunity of DP. Indeed, the noise is not added at the end, over the infinitely private sign of the age, rather, it is added before the computation of the sign, inside the mechanism and not afterwards. Thus, the noise addition cannot be considered as a post-processing.

3 - Combining homomorphic encryption and differential privacy in federated learning

Abstract Recent works have investigated the relevance and practicality of using techniques such as Differential Privacy (DP) or Homomorphic Encryption (HE) to strengthen training data privacy in the context of Federated Learning protocols. As these two techniques cover different sources of confidentiality threats (other clients/end-users for the former, aggregation server for the latter), there is a need to consistently combine them in order to bridge the gap towards more realistic deployment scenarios. In this paper, we achieve that goal by means of a novel stochastic quantisation operator which allows us to establish DP guarantees when the noise is both quantised and bounded due to the use of HE. The paper is concluded by experiments on the FEMNIST dataset which show that the precision required to get state-of-the-art privacy/utility trade-off (which directly impacts HE parameters and, hence, HE operations performances) results in a 3.6% computation time overhead imputable to HE calculations, for the whole training of a 500k parameters model.

N.B.: This chapter is the reproduction of the article *Protecting data from all parties: Combining fhe and dp in federated learning*, joint work with Renaud Sirdey, Oana Stan and Cédric Gouy-Pailler, to be submitted (preprint available on [84]).

3.1 . Introduction

In 2016, McMahan et al. proposed a new paradigm of collaborative learning that they called Federated Learning (FL) [126]. This collaborative method allows to train a machine learning algorithm across multiple actors without exchanging data samples. Instead, there are local trainings based on local data samples and an exchange of the model's parameters in order to generate a global model. The most classical federated learning setting relies on a central aggregation server which coordinates the other participating actors (also called clients) and aggregates the model updates. Along with the reduction of communication load and the parallelism it allows, a claimed key advantage is the protection of data due to the fact that each client keeps its own data locally. However, although FL gives some protection to the data with regards to the server, it gives rise to a new type of potential adversaries - the other clients. Several attacks that take advantage of this new threat were proposed in [96, 130].

3.1.1 . Our contribution

The contribution of this paper is an approach to consistently combine countermeasures of different natures, namely Differential Privacy and Homomorphic Encryption, with the aim to enable the integration of both in more secure FL frameworks. Indeed, the above-mentioned attacks on the training data can be mitigated via DP, either if they come from the other participants of the training process or from the end-users of the model. Other potential threats

come from the central aggregation server. Homomorphic encryption can then allow to mitigate these later threats without any communication between the clients: the clients send encrypted information to the server which will do the necessary computations in the encrypted domain, without seeing either the sent information or the result of its computations. In order to consistently articulate DP and HE we introduce a new stochastic quantisation operator based on the Poisson distribution. This operator behaves as if it was applied as post-processing of a Gaussian mechanism, keeping the DP guarantees of this standard mechanism unchanged without any supplementary analysis and allowing to seamlessly get rid of the quantisation issue due to the use of HE. Note that this harmless quantisation technique is of independent interest in a context of communication constraints and DP requirements, even without use of cryptographic techniques.

An illustrative application scenario could be in the medical field. We may consider several hospitals that own medical data from their patients and wish to collaborate in order to train a global model that would detect a certain disease. In many countries, patient data are sensitive and the hospitals are not allowed to share them with other hospitals. A solution is to use a Federated Learning protocol (*e.g.* from an institutional entity) but without the hospitals disclosing their data to the aggregation server. Note that the parameters we used for our experiments on FEMNIST (see Section 3.5) make our privacy building blocks scale at a level which can be for example compatible with the number of medical facilities in a reasonably large country.

The paper is organised as follows. A review of the literature on the issues of data privacy in a FL context follows this introduction in Section 3.2. Then, Section 3.3 provides the technical prerequisites on HE schemes necessary to understand our method, that we thoroughly explain in Section 3.4. The results of the experiments that we ran to illustrate the feasibility of our solution are presented in Section 3.5, before concluding remarks and perspectives for further work (Section 3.6).

3.2 . Related work

3.2.1 . Differentially private federated learning

Due to the additional threats from the other clients, DP has been quite popular in collaborative and especially FL frameworks. For instance, McMahan et al. [127] trained a next word prediction algorithm in a federated way, using text data from users' smartphones, protected by DP. Nevertheless, this work is hardly comparable to ours because the task for which the model is trained is very different from our targeted classification tasks (*e.g.* FEMNIST in our experiments), inducing very different learning parameters and thus a quite different privacy challenge. Closer to our work is the one from [77] which presents a differentially private FL process tested on MNIST, yet a notably easier task than FEMNIST, with a quite high privacy cost ($\epsilon = 8$ and $\delta = 10^{-3}, 10^{-5}, 10^{-6}$). Shokri et al. [166] also experimented a differentially private distributed learning setting - similar to a FL setting due to the use of distributed stochastic gradient descent - on MNIST, and SVHN, another famous image dataset. In [156], Sabater et al. ensure privacy and utility of their distributed learning setting as long as each participant communicates

with only a logarithmic number of other participants. Abadi et al. [2] introduced the moments accountant method, a useful tool for our DP analysis, that enables to keep track of the privacy cost more tightly than the traditional composition theorem when many calls to the database are done, typically for training a deep neural network.

3.2.2 . Cryptographic primitives for federated learning

Most of the works applying HE to machine learning models focus on the inference stage (CryptoNets [78], TAPAS [160], NED [91]) and not on the training stage. The first papers on privacy-preserving machine learning training focused on a centralised setting where all data are outsourced and where the models are only linear [19, 38]. When it comes to non-linear models, the few approaches that ran a complete centralised training of neural networks on encrypted data have impractical performances or huge cryptographic parameters [120].

While some authors propose solutions in the case of multi-servers either for clustering or regression, many methods employing HE have been recently proposed for a collaborative learning task with no central server. They mostly apply on linear models [117, 194] and, more recently, Sav et al. focused on neural networks [161]. As far as centralised FL is concerned, there are a few recent papers proposing the use of HE to protect the clients' data from the server [149, 189, 192], where [149] and [189] are only theoretical.

More general than HE, multi-party computation protocols for the problem of secure aggregation allow several agents to collaborate and compute a function on their data such that each agent knows no more than its own input and, if requested, the output, and learns nothing about the other agents' inputs. The combination of the high-communication costs of the multi-party computation and the inherent distributed nature of FL makes FL methods based on multi-party computation or ad hoc approaches (*e.g.* [25, 134]) difficult to implement efficiently. Among these approaches, [101] interestingly makes use of DP techniques to further protect the private data.

The work from [171] combines DP with secure aggregation for federated learning and proposes two versions of the same protocol: one for the case of trusted (semi-honest) server and another, more advanced, for malicious server. The aggregation protocol is based on the security of LWE problem and on the MPC protocol of Packed Shamir secret sharing scheme [72]. The DP guarantees are ensured by the error induced by LWE encryption. As for the case of malicious server, they extend the protocol with the Benaloh's verifiable secret scheme [18]. We remark however that the communication and computation complexities are quite high.

3.2.3 . Quantisation and differential privacy

The principal focus and key contribution of our paper deals with the interference between DP and HE. The main issue induced by this interference is that the domain of the messages to be encrypted (which are the noised updates in our work) has to be discrete and bounded (and encoded with as few bits as possible for better efficiency of the FHE computations). For other reasons (unrelated to encryption), some authors have studied the possibility of using discrete noises for differential privacy.

For instance, the authors of [6] propose a secure and communication efficient distributed learning framework. They perform the DP analysis of their learning mechanism using a bino-

mial noise because the effect of quantisation on the Gaussian mechanism is unclear, especially after aggregation if the noise is generated in a distributed way. The analysis is quite involved and only provides DP bounds for the multidimensional binomial mechanism for one round of learning. Indeed, the moments accountant method is not easily applicable to the binomial distribution. Moreover, the presented DP guarantee is worse than the Gaussian mechanism's one and needs the quantisation scale to tend to zero (and hence the communication cost to infinity) to approach it.

In [111], Koskela et al. present a privacy accountant for discrete-valued mechanisms for non-adaptive queries using privacy loss distribution formalism and Fast Fourier Transform. In particular, they give DP guarantees for the binomial mechanism in one dimension and extend them in the multidimensional case but with quite demanding constraints that compel them to brutally approximate the gradients by their sign in their experiments. Cannone et al. [34] introduce the discrete Gaussian mechanism and studied its DP guarantees that scale well with composition, even in the multivariate case. Nevertheless, contrary to binomial noise, discrete Gaussian noise is not bounded as required for our framework. More critically, the discrete Gaussian distribution is not stable by addition, thus precluding its direct use in a context of distributively generated noise that a collaborative learning task with untrusted server requires (see Section 3.4.1).

In [5, 106], the authors propose federated learning protocols protected by DP and secure aggregation (which requires discrete and bounded values, as HE, but needs communication before learning as mentioned in 3.2.2). These works respectively use the discrete Gaussian mechanism and the Skellam mechanism to ensure DP. At the cost of a careful DP analysis, they show that, for fine enough quantisation scale, their DP guarantees approach the Gaussian mechanism's ones. Our work proposes a much simpler way to obtain the very same guarantee as the Gaussian mechanism, without needing to constrain the quantisation scale and with much simpler mathematical analysis. Moreover, the two previous works have to make use of conditional randomised rounding to ensure that the rounding of the unnoised values does not increase their norm too much. Since we perform quantisation after noising with a quantisation that can be viewed (from the DP perspective) as a post-processing (see Section 3.4.3), we do not have such an issue.

3.3 . Preliminaries on homomorphic encryption schemes

A common characteristic of some of the most popular HE schemes (BGV [31], BFV [69]) is their plaintext domain defined over the ring $R_t = R/tR$ with $R = \mathbb{Z}[x]/f(x)$ the polynomial ring modulo the function f and the integer $t \geq 2$. The typical choice for f is $(X^n + 1)$ with n a power of 2. As such, before encryption, each message has to be encoded as a plaintext consisting in a polynomial of degree smaller than n with integer coefficients from the range (0, t-1), and all operations over individual elements are performed modulo $(X^n + 1)$, and modulo t. The ciphertext space for these schemes is $R_q = R/qR$.

Moreover, a lot of HE schemes, like BFV that we use in our experiments (Section 3.5), offer a *batching* capability by which multiple cleartexts can be packed in one ciphertext resulting in

SIMD (Single Instructions Multiple Data) homomorphic operations i.e.,

$$\mathsf{Enc}(m_1, ..., m_\kappa) \oplus \mathsf{Enc}(m'_1, ..., m'_\kappa) = \mathsf{Enc}(m_1 + m'_1, ..., m_\kappa + m'_\kappa)$$

(and similarly so for $\otimes$). Typically, several hundreds such slots are available (for BFV, the maximal number of slots coincides with n), which often allows to significantly speed up encrypted domain calculations. In order to apply batching for BFV, t has to be prime and $t = 1\ mod[2n]$.

3.4 . An illustrative privacy-preserving federated learning framework

In this section, we present a simple FL framework (depicted in Figure 3.1). As this paper focuses on quantisation techniques suitable to derive DP guarantees when an FL aggregation server works in the FHE domain, this framework is meant to be illustrative. In particular, it should be emphasised that any realistic secure FL framework would depend on additional (mostly off-the-shelf) countermeasures (*e.g.* strong authentication of all involved parties, confidentiality and integrity of all messages, realistic key generation and agreement protocols, etc.) that are beyond the scope of the present paper. So, in our simplified framework, the entities that take part of the learning process are the server and the clients, that own a private FHE decryption key (note that the techniques presented in this paper are compliant with more recent multikey FHE approaches such as [145]) and among which participants are sampled at each round. After a common initialisation of the model, the process at each iteration is the following:

- the server sends the encrypted current model parameters to the participants;
- each participant decrypts the parameters thanks to the decryption key (except at the first iteration when the parameters may not be encrypted);
- each participant performs a stochastic gradient descent of the loss function using its local data;
- each participant applies on the computed updates the successive transformations required by DP and HE (clipping, noising and quantisation);
- each participant encrypts and sends the transformed updates to the server;
- the server aggregates (averages) the updates in the encrypted domain.

On the need to combine DP and HE: Per se, the simple architecture described above is limited in terms of threat model to honest-but-curious non-colluding actors. This is true of the clients which are then assumed to properly perform their task but attempt to infer as much information as possible on the training data of other clients from what they legitimately see. This is also true of the end-users (the clients themselves or the entities which do not participate in the training process and see only the final model) and of the server. Under this very basic model, DP mitigates confidentiality threats (on the clients' training data) from the clients and end-users, while HE mitigates confidentiality threats (on the clients' training data) from the server. This exemplifies the need for being able to articulate both techniques to address even the most basic threat models emerging from the previous illustrative framework.

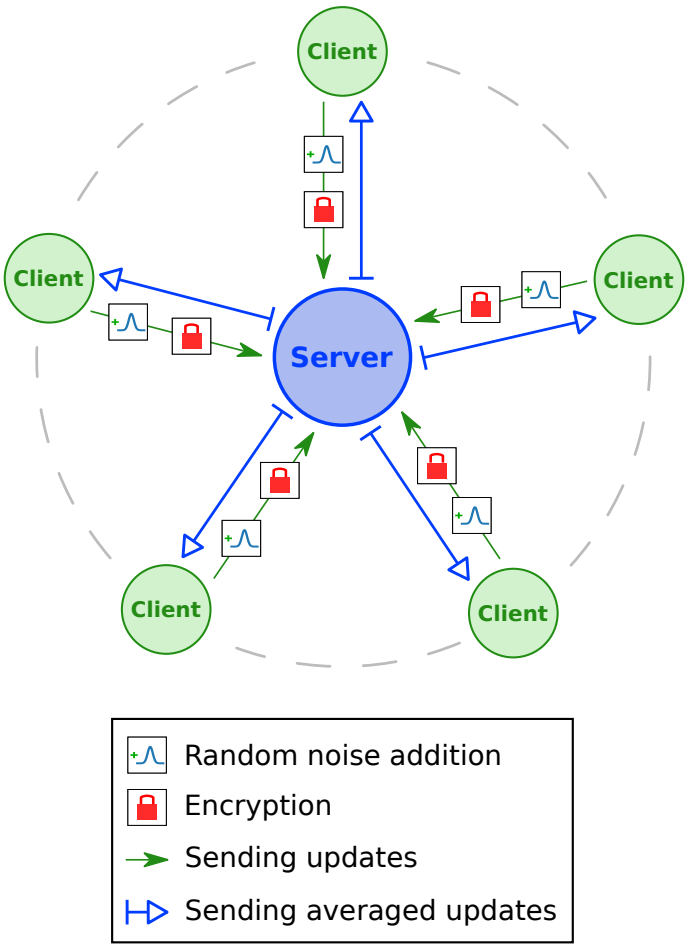

Figure 3.1: An illustrative baseline secure federated learning architecture.

3.4.1 . Distributed noise generation

When willing to protect the training data by DP in a collaborative learning process, having the participants generate the noise in a distributed way [83,156] rather than to rely on the server to do so is desirable to mitigate a server that would communicate the noise to some clients or end-users and then break the DP guarantees. Of course this would not be *stricto sensu* needed for the most basic threat models dealing only with honest-but-curious non colluding adversaries. In that case, the central noise would be generated in the clear domain and homomorphically added to the aggregated updates, and quantisation would not cause any difficulty.

The distributed noise generation, that we here adopt, is especially practical when one wants the resulting noise to follow a Gaussian distribution, since this distribution is stable by addition. The participants simply need to generate Gaussian noises with well-chosen variances. However, DP in a FL context still requires adaptations of the FL process:

1. clipping the updates in L2-norm with the clipping bound S (*i.e.* substituting participant k's vector of updates u_k by $\min\left(1, \frac{S}{\|u_k\|_2}\right).u_k$) to bound the sensitivity (i.e. the impact of changing from one dataset to an adjacent one) since unbounded sensitivity is incompatible with any DP guarantee;

2. adding noise to the gradients (*e.g.* Gaussian noise);

3. fix all the coefficients of the mean to $\frac{1}{K}$, independently of the size of the participant's dataset, to bound the sensitivity more easily.

3.4.2 . Problem of the limited number of bits and first approaches

In our scenario, the information sent by the participants to the server is encrypted via HE. Since floating-point homomorphic calculations (although in principle possible) are prohibitively costly, we have to switch to a fixed-point representation while avoiding using too many bits (as the cost of FHE calculations will increase, due to several factors discussed in Section 3.5, with the number of bits required to represent the plaintexts). This means that, unlike the usual case where the noise is represented by a double-precision float (i.e. finite but very fine precision) and where we make the assumption that it perfectly follows the desired distribution, we here have to explicitly take into account bounds on the noise and quantisation of this noise (and of the updates themselves). However, if we round the noised updates in a traditional way (scaling and rounding to the nearest integer), the aggregated noise is not Gaussian any more, but it is a sum of noises that follow rounded Gaussian distributions. Unfortunately, the distribution of a sum of rounded Gaussian variables does not have a simple expression (in particular, it is not a rounded Gaussian distribution as shown in Appendix E)[1].

A naive approach would be to try to compare the complicated distribution of a sum of rounded Gaussians with the one of a sum of perfect Gaussians i.e. a Gaussian. Indeed, if the quantisation scale is fine enough, the sum of rounded Gaussians should intuitively be close to a Gaussian. The final privacy cost should be the sum of the one of a classical Gaussian mechanism

[1]If such a distribution were easily dealt with, the issue of boundedness could be addressed by considering the modulo operation induced by encryption, like in 3.4.5.

and a hopefully small additional privacy cost due to the approximation. Nevertheless, this analysis is quite involved and may result in overestimated DP bounds.

Another idea is to use binomial noise instead of Gaussian noise as in [6,111] but, as explained in Section 3.2.3, due to the fact that it is not rotation invariant, the binomial distribution is hard to use in multidimensional problems and the moments accountant method does not apply directly to it, as it does for the Gaussian distribution. Besides, the DP guarantee obtained in [6] with an involved mathematical analysis needs the quantisation scale to tend to 0 to get close to the Gaussian mechanism's guarantee while communication constraints precisely require this scale to be large.

We will now show that, under natural practical assumptions and with the use of a novel specific quantisation function, we may significantly simplify the DP analysis, with no privacy loss compared to the Gaussian mechanism. Most importantly, our DP guarantee does not depend on the quantisation scale, *keeping us free from the trade-off between privacy and communication* faced in [5, 6, 106].

3.4.3 . Poisson quantisation

We here propose a new probabilistic quantisation operator that commutes with the sum[2], and is therefore harmless for the DP guarantee of the mechanism. In the following, $\mathcal{P}(\lambda)$ denotes the Poisson law of parameter $\lambda \in \mathbb{R}_+^*$ whose support is $\mathbb{N}$ and whose probability mass function is $k \in \mathbb{N} \mapsto \frac{\lambda^k}{k!} e^{-\lambda}$. We fix the quantisation scale $s \in \mathbb{R}_+^*$ and the dimension $d \in \mathbb{N}^*$ of the problem (the number of parameters of the model in our case).

Definition 8. *Let $\mu \in s\mathbb{Z}$. We define the probabilistic function*

$$Q_{s,\mu} \colon x \in]\mu; +\infty[\mapsto sY + \mu$$

where $Y \sim \mathcal{P}\left(\frac{x-\mu}{s}\right)$. We call it the Poisson quantisation *of scale s and offset μ.*

Similarly, we define

$$Q_{s,\mu} \colon x = \left(x^{(i)}\right)_{i \in [\![1;d]\!]} \in]\mu; +\infty[^d \mapsto \left(Q_{s,\mu}\left(x^{(i)}\right)\right)_{i \in [\![1;d]\!]}.$$

Given $\mu \in s\mathbb{Z}$, for all $x \in]\mu; +\infty[^d$, $Q_{s,\mu}(x)$'s support is $(s\mathbb{Z})^d$ and its mean is equal to x so we can actually consider the Poisson quantisation as a function of probabilistic quantisation. Proposition 11 shows that the Poisson quantisation on the terms of a sum can be considered as a post-processing on the sum.

Proposition 11. *Let $m \in \mathbb{N}^*$, $x_1, \dots, x_m \in \mathbb{R}$. Let $\mu \in s\mathbb{Z}$ such that $\mu < \min\{x_i | i \in [\![1;m]\!]\}$. $Q_{s,m\mu}\left(\sum_i^m x_i\right)$ has the same distribution as $\sum_i^m Q_{s,\mu}(x_i)$.*

Proof. $\sum_i^m Q_{s,\mu}(x_i) \sim \sum_i^m (sY_i + \mu) = s\sum_i^m Y_i + m\mu$ where, for all $i \in [\![1;m]\!]$, $Y_i \sim \mathcal{P}(\frac{x_i-\mu}{s})$. By stability of the Poisson law by addition, we know that $\sum_i^m Y_i \sim \mathcal{P}\left(\sum_i^m \frac{x_i-\mu}{s}\right) = \mathcal{P}\left(\frac{\sum_i^m x_i - m\mu}{s}\right)$. We then directly get the result. □

[2]Commutativity must be understood in a large sense, as the offset parameter of the quantisation changes depending on the order of the operators.

Proposition 11 together with Proposition 1 enables us to conclude that Poisson quantisation has no influence on the DP guarantee. Indeed, the output distribution is the same as if we had applied the Poisson quantisation after the aggregation of the *continuously* noised updates. Since adding continuous Gaussian noises distributively on the updates and add them afterwards amounts to add a Gaussian noise to the sum of the unnoised updates, the Poisson quantisation acts as if it was applied on top of the Gaussian mechanism. *Hence, the huge advantage of our Poisson quantisation operator is that it allows to reduce the DP analysis back to the vanilla analysis of the Gaussian mechanism* (see Section 3.4.6).

Note that, since Poisson quantisation is probabilistic, it might harm the accuracy of the model. Given $\mu \in s\mathbb{Z}$ and $x \in]\mu; +\infty[$, the variance of $Q_{s,\mu}(x)$ is $s^2\frac{x-\mu}{s} = s(x-\mu)$. For a small enough s, this variance is very small since x is bounded, and there is actually no impact on accuracy in our experiments (Section 3.5).

An important point to notice is that Poisson quantisation implies that the values to quantise have an *a priori* common lower bound (otherwise the sum of the quantised values may depend on these values and not only on their sum). In our case, these values are the noised updates. The updates are already bounded by the clipping: as we will see in Section 3.4.6, this clipping constrains the L2-norm of the updates (considered as vectors of the updates of all parameters) and thus it also constrains the absolute value of each parameter update. As for the noises, the following section shows we can consider that the noises have a common lower bound in practice.

3.4.4 . Bounded Gaussian noises

The most common algorithms to sample from a Gaussian distribution are Box-Muller transform in its Cartesian and polar forms ([29, 110]) and the ziggurat algorithm ([124]). As they rely on a source of uniform randomness, we here show that all these algorithms actually generate values whose range have bounds which are way smaller than the range of double-precision floats.

Box-Muller transform (Cartesian form) [29]: The Cartesian form of the Box-Muller transform samples two independent uniform random variables U_1 and U_2 in $[0; 1]$. The random variables $Z_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2)$ and $Z_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$ are independent and follow a standard normal distribution (standard deviation 1 and mean 0). Since the function $\cos$ and $\sin$ are bounded by -1 and 1, we see that the maximum absolute value of Z_1 and Z_2 is reached for the minimum value of U_1 which is 2^{-nBits} where $nBits$ is the number of bits used to represent an integer. For 64 bits, we get $\sqrt{-2\log(2^{-64})} \approx 9.42$.

Box-Muller transform (polar form) [110]: The polar form of Box-Muller transform samples two independent uniform random variables U_1 and U_2 in $[-1; 1]$ and calculates $s = U_1^2 + U_2^2$. The random variables $Z_1 = \sqrt{-2\log(s)}\frac{U_1}{\sqrt{s}}$ and $Z_2 = \sqrt{-2\log(s)}\frac{U_2}{\sqrt{s}}$ are independent and follow a standard normal distribution. In this case, $\frac{U_1}{\sqrt{s}}$ and $\frac{U_2}{\sqrt{s}}$ always belong to $[-1; 1]$ hence the maximum absolute value reached by Z_1 and Z_2 is $\sqrt{-2\log(s_{min})}$ where s_{min} is the minimum value possibly reached by s, namely $s_{min} = (2^{-nBits})^2 + (2^{-nBits})^2 = 2^{-2nBits+1}$. For a 64-bit processor, this gives $\sqrt{-2\log(2^{-127})} \approx 13.27$.

Ziggurat algorithm [124]: The ziggurat algorithm applies to monotonically decreasing

probability distributions and extends to symmetric unimodal distributions like the normal one by randomly choosing on which side of the mode the sampled value will fall. The algorithm works by covering the distribution by stacked rectangular regions of same area. What matters for our problem of finding the bound of the sampling process is only the tail rectangle. In the rare case where the value did not fall into one of the other rectangles, a fallback algorithm is used to sample the value from the tail. Let x_{tail} be the abscissa of the right side of the last rectangular region before the tail. The fallback algorithm for the normal distribution samples two independent uniform random variables U_1 and U_2 from $[0; 1]$ and defines $x = -\log(U_1)/x_{tail}$, $y = -\log(U_2)$. It then tests if $2y > x^2$ and returns $x + x_{tail}$ if yes. Otherwise, it restarts with two new samples U_1 and U_2. Thus, the biggest value in the monotonic case, also the biggest absolute value in the symmetric case, is $-\log(2^{-nBits})/x_{tail} + x_{tail}$. In [124], Marsaglia and Tsang calculate that, for 255 rectangles, $x_{tail} \approx 3.65$. For 64 bits, we then get the bound 15.81.

These "artificial" bounds, that we cannot avoid in practice anyway, are justified by the very low probability of a draw outside them: less than 10^{-20} for the lowest bound, 9.42, and less than 10^{-55} for the highest, 15.81. To get a sample from an arbitrary normal distribution, it suffices to scale the sample of the standard normal distribution by the wanted standard deviation and then add the wanted mean. This discussion allows us to exhibit a lower bound for the Gaussian noises and, the unnoised updates being bounded by the clipping, for the noised updates. As a consequence we can apply Poisson quantisation.

Note that the distributions followed by the outcomes of those popular sampling algorithms are almost invariably (and implicitly) considered in the literature as perfect normal distributions. We will make the same assumption here and deem that, if we can represent the output of these algorithms without any loss of information (perfectly represent all double-precision floats in $[-16; 16]$ for example) then the limited number of bits of the messages does not have any impact on the span of the noise. We can thus ignore the boundedness of the noise in the DP analysis.

3.4.5 . The problem of the unbounded Poisson distribution is not a problem

A drawback of Poisson quantisation is that it is not bounded, while the cryptosystem only works on a finite set of values. Indeed, even if the Gaussian noises are bounded by the practical limitations of their sampling algorithm, the quantised noised updates are not. However, we show in this section that this is actually not a problem for our method.

A first argument to address the issue of the theoretically infinite range of the Poisson distribution would be to study Poisson sampling algorithms and find a practical bound, like we did for the normal distribution (Section 3.4.4). Nevertheless, this does not seem that straightforward for the Poisson case[3].

Rather, let us see what happens if the Poisson sample falls out of the bounds imposed by the cryptosystem plaintext domain (i.e. exceeds the plaintext modulus). At encryption, a modulo operation will automatically be applied to the value. The same modulo operation will

[3]Although traditional sampling algorithms only generate very large size outcomes with very low probability and very large computation time.

be performed on the aggregated updates on the server side. Observation 1 shows that these two modulo operations amount to a single modulo operation on the sum of the updates, which constitutes a post-processing on this sum and, as such, does not affect the DP analysis.

Observation 1. *Let* $(x_i)_{i\in[\![1;K]\!]} \in \mathbb{Z}^K$, $N \in \mathbb{N}^*$.

$$\sum_{i=1}^{K}(x_i \mod N) \mod N = \sum_{i=1}^{K} x_i \mod N.$$

Recall that only integer values can be manipulated in the encrypted domain. This implies that the quantised noised updates are multiplied by the inverse quantisation scale $\frac{1}{s}$ before being encrypted, and that the participants rescale the averaged updates by s once received from the server at the next round.

Let us now consider the influence of this modulo operations on the accuracy of the model. First of all, the result of the Poisson quantisation may be non-positive due to the negative offset μ and thus may fall out of $[0, \ldots, N-1]$, where N denotes the plaintext modulus. To avoid this situation, we make the participants send the quantised updates without adding the (potentially non-positive) offset μ. When they receive the averaged updates from the server, they just have to add μ to them to get the actual averaged updates. The second case is encountered when a sample exceeds the plaintext modulus. Nevertheless, this event is very rare if the modulus is big enough. With the parameters we use (Section 3.5), we show, using Chebyshev's inequality, that the probability for a quantised gradient to exceed the plaintext modulus is lower than 1.01×10^{-9}. Similarly, the probability that the aggregation (sum) of the K quantised gradient exceeds the plaintext modulus is lower than 1.61×10^{-5}, to compare to the number $486,654$ of parameters. In any case, our experiments prove that this has no practical influence on the model accuracy.

3.4.6 . DP analysis of the Gaussian mechanism

According to the discussion above, our mechanism has the same DP guarantee as a mechanism where true unbounded and continuous Gaussian noise is added by the server *after* aggregation, a.k.a. the Gaussian mechanism. The noise introduced as a side-effect by Poisson quantisation may even improve the privacy but, for simplicity, we consider it as banal post-processing and do not take it into account in the DP analysis. Hence, the DP analysis reduces to the Gaussian mechanism's analysis. As explained in 2.1.4 and pretty much like in [77] for instance, we use the moments accountant [2] to compose privacy costs in an efficient way across the multiple learning rounds.

Formally, given $\sigma \in \mathbb{R}_+^*$ the standard deviation for the aggregated noise and $S \in \mathbb{R}_+^*$ the clipping bound in L2-norm, let us consider the two density functions corresponding to two adjacent databases respectively not containing and containing the adversary's target client:

$$f_1 \colon x \in \mathbb{R} \mapsto \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

and

$$f_2 \colon x \in \mathbb{R} \mapsto \frac{1-q}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} + \frac{q}{\sigma\sqrt{2\pi}} e^{-\frac{(x-2S)^2}{2\sigma^2}}$$

where $q = \frac{K}{M}$ is the fraction of participants by round i.e. the probability of a given client being chosen to participate in a given round. Since $q < 1$, we get privacy amplification by *subsampling*. Without loss of generality since the integral for the moments accountants computation is on whole $\mathbb{R}$, f_1 is defined with mean 0. Note that the part corresponding to the event whereby the target client is chosen as participant has an offset of $2S$ rather than S because, if the absolute values are constrained by the clipping bound S, the actual span of the values is $2S$. As a result, changing the participant may modify the updates by $2S$.

The moments accountant of order $l \in \mathbb{R}_+^*$ corresponding to a single query to the private database, i.e. a single learning round, is:

$$\alpha(l) = \max \left[\int_{\mathbb{R}} \left(\frac{f_1(x)}{f_2(x)} \right)^l f_1(x)dx, \int_{\mathbb{R}} \left(\frac{f_2(x)}{f_1(x)} \right)^l f_2(x)dx \right].$$

The total moments accountant of the learning process is $\alpha_{total} \leq T \max_{l \in \mathbb{R}_+^*} \alpha(l)$ where T is the number of learning rounds. In practice, we compute the max for l being an integer varying in $[1, \ldots, 20]$. Finally, we apply Theorem 2 to derive the DP guarantee ϵ from α_{total}, $\delta << K$ being chosen (in our experiments, we took $\delta = 10^{-5}$).

Privacy cost from the point of view of a participant

For a comprehensive analysis, one must not forget that, from the point of view of a participant k, the noise generated by k does not participate in the privatisation process[4]. Hence, we must take into account only the other participants' noises. The individual noises added by the participants are calibrated such that their sum has a certain standard deviation σ i.e. these individual noises have standard deviation $\frac{\sigma}{\sqrt{K}}$, K being the number of participants in each round. As a result, the DP guarantee from the point of view of a single participant must be computed by substituting σ by $\frac{\sqrt{K-1}}{\sqrt{K}}\sigma$, which has an insignificant influence if K is large (1000 in our experiments). Note that this is still quite conservative as it assumes that the considered participant may participate to all training rounds.

We can also interestingly extend our threat model in a straightforward way by considering that some clients may collude and share their noises with each other, quite like in [83]. From the point of view of a colluding client, the noise added by all the colluding clients would be known and it would therefore not participate in the DP protection of the data. The aggregated noise would then have a standard deviation of $\sqrt{1-\chi}\sigma$ and this would result in a degraded DP guarantee, obtained by substituting σ by $\sqrt{1-\chi}\sigma$ where χ is the ratio of colluding participants. This modification of the DP guarantees also applies in the case of some clients dropping out. Figure 3.3 illustrates the loss of confidentiality due to collusion of clients.

3.4.7 . Homomorphic encryption protects the data (and the model) against the server

While considering our learning framework protected by a distributed noising, it may not be clear why the framework ever needs to make use of cryptography. Indeed, the server receives

[4]For instance, if one knows the noise that was added to a value, one just has to remove this noise from the noised value to get the initial value.

the updates from the participants after they have been noised. However, each individual noise has been calibrated such that the *aggregated* noise will obfuscate the sensitive information of a specific participant. If σ is the standard deviation necessary to hide the data of one participant, the standard deviation of each individual noise is $\frac{\sigma}{\sqrt{K}}$. However, since without HE the server would see each individual noise updates *before* aggregation, the individual noise should be equal to σ if it were to protect the updates from the server. Such a setting is referred to as *local DP* in the literature. Yet, in our case, this would result in an aggregated standard deviation of $\sqrt{K}\sigma$ (for the sum, or $\frac{\sigma}{\sqrt{K}}$ for the average) which would heavily harm the utility of the averaged updates and thus the accuracy of the model.

In terms of concrete HE, the fact that we are considering the simple FederatedAveraging operator allows us to spare much computation time by using additive-only schemes such as in [122] where the Paillier cryptosystem is used with batching. In the experimental results reported in the next section we have used the BFV cryptosystem which allows for more massive batching and, as such, results in much lower (amortised) overheads. Additionally, one key contribution of [122] was to associate Paillier-based homomorphic calculations to Verifiable Computing (VC) techniques (*e.g.* [71]) to further extend the server threat model beyond the honest-but-curious one and bring execution integrity, as [121] did with BFV scheme. However, these works lacked DP. Indeed, adding the DP noise on the server requires a tag that can only be generated with knowledge of the VC scheme secret key (i.e. by a client), meaning, in the Federated Learning context, that at least one of the clients would have knowledge of the total noise added (resulting in a collapse of the DP guarantee regarding this client, even when that knowledge is uncertain). We could actually imagine that the server generates K noises that sum to the required noise and send each of them to a participant for tag generation but this would double the communication cost. Hence, associating DP with VC for server-side computation integrity maintaining a reasonable communication cost requires a distributed noise generation as provided in this paper. As such, the noise generation technique proposed in this work is directly applicable to setups where homomorphic calculations are paired with VC techniques.

As a very interesting side-effect, the HE layer also hides the model parameters from the server throughout the training. This may be valuable when the clients want to keep their model private, or give only a black-box access to it, either for privacy or economic reasons (cf. machine learning as a service).

3.5 . Experimental results

To prove the practicality of the combination of our Poisson quantisation technique with HE, we performed experiments that enable us to evaluate training performance in terms of accuracy, precision requirements and computation time. We chose the Federated Extended MNIST (FEMNIST) dataset[5] to run the experiments. The extended version of MNIST contains 62 classes (digits, upper and lower letters) of hand-written characters from 3,596 writers and comes with the writer id. FEMNIST, the federated version, was built by partitioning the data based on the writer [33]. The network architecture is the same as in [122]: a standard CNN

[5]Dataset available at https://www.nist.gov/itl/products-and-services/emnist-dataset

composed of two convolution layers (respectively with $5 * 5$ kernel size and 128 channels, and with $3 * 3$ kernel size and 64 channels, each followed with $2 * 2$ max pooling), a fully connected layer with 128 units and ReLu activation, and a final softmax output layer (486,654 parameters).

Table 3.1 shows the influence on the model accuracy of the adaptations necessary to ensure DP. Starting from a non-DP baseline from the state of the art [122], we successively modified parameters of the framework, each of these modifications being required by the DP analysis[6]. The successive steps are:

- reduce the number of learning rounds from 200 to 100, hence reducing the amount of queries to the clients' sensitive datasets: as shown in Table 3.1, this has a very mild influence on the model accuracy whereas, for smaller number of learning rounds, the accuracy starts decreasing more significantly
- increase the total number M of clients (to 3596, the total number of writers for FEMNIST) and the number K of participants per round to 1000. This has two advantages. Firstly, we can make the ratio $q = \frac{K}{M}$ smaller, decreasing the probability of a target client participating at a given round and thus the probability of this target releasing any information during this round. Secondly, the absolute value K is greater, so that the information of the target participant is more diluted in the averaged updates. In practice, the experiments show that, with a fixed distortion ratio $\frac{\sigma}{K}$, which gives roughly the same model accuracy, the DP guarantee ϵ decreases when K increases. We then chose $K = 1000$, in our opinion the largest reasonable value so that a substantial ratio of the clients can stay idle at each round. The impact on the accuracy of the increasing of M and K is due to the larger number of writers, inducing a higher variety in the training samples (non-i.i.d. across the different writers) which makes the classification task more complex.
- assign the same coefficient $\frac{1}{K}$ to all the participants in the weighted average (rather than the proportion $\frac{n_k}{n}$ of training samples owned by participant k) so that the sensitivity of the average for every participant is $\frac{S}{K}$ rather than $\max_{k \in [\![1;K]\!]} \frac{n_k}{n} S$, where $\max_{k \in [\![1;K]\!]} \frac{n_k}{n}$ may be much larger than $\frac{1}{K}$
- clip updates with clipping bound S to ensure finite sensitivity (we took $S = 1$ which has a mild impact on accuracy and allows for good DP guarantees)
- on the participant side, quantify the noised updates via Poisson quantisation
- apply a modulo operation on the noised updates on the participant side, and on their sum on the server side, which is automatically done by encryption
- add the Gaussian noise necessary to make the learning process differentially private. We chose $\sigma = 6$ for the total noise because it gives a good trade-off between privacy and model accuracy as shown in Figure 3.2.

[6]Note that the order in which we made these successive adaptations does not correspond to the order in which they are executed in the learning workflow.

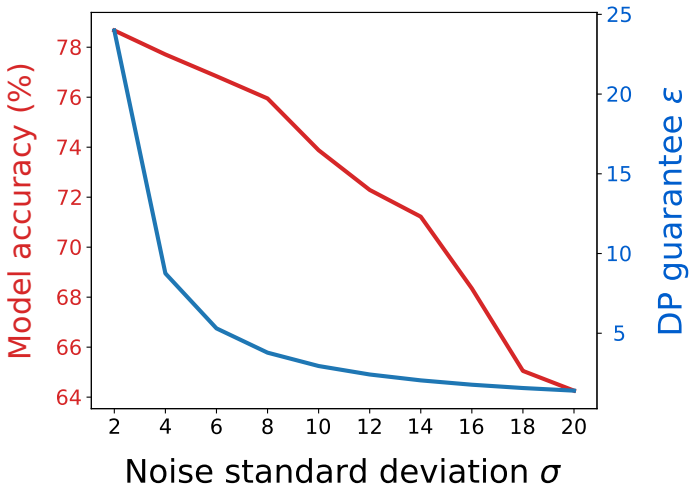

Figure 3.2: Model accuracy and DP guarantee vs noise standard deviation ($\delta = 10^{-5}$)

We fixed the scale for Poisson quantisation to 10^{-4}, as in [122], since it does not much affect the accuracy. We used as common lower bound μ of the Gaussian noises the lower bound from the ziggurat algorithm with 255 rectangles, i.e. -15.81 (see Section 3.4.4), multiplied by the standard deviation of the distribution. This lower bound is greater (in absolute value) and then more conservative than the lower bounds of the two other sampling algorithms we considered. Moreover and quite importantly, ziggurat algorithm with 255 rectangles is actually the algorithm chosen by the numpy library we used.

The whole training process is (ϵ, δ)-differentially private, with $\epsilon = 5.31$ and $\delta = 10^{-5}$. Actually, for $\delta = 10^{-5}$, $\epsilon = 5.306$ for an end-user which is not a participant and $\epsilon = 5.309$ for a participant (see discussion at the end of Section 3.4.6). More widely, Figure 3.3 represents the privacy cost, from the point of view of a colluding participant i, as a function of the ratio of participants who collude with i. As expected, we see that the privacy cost increases smoothly with the ratio of colluding participants and, up to say 20% of colluding participants (i.e. 200 colluding teachers) the privacy cost remains reasonably close to the one in the non-colluding case.

These DP guarantees together with the model accuracy of 76.84% give us the same privacy/utility trade-off as [5, 106] got with secure aggregation. Nevertheless, if communication is a critical issue, we may use a greater quantisation scale, at the expense of accuracy, but this would not harm the DP guarantee, contrary to [5, 106]. Interestingly, we experimentally notice that the quantisation and the modulo operation have no influence on the accuracy: the model trained with noise but without quantisation or modulo operation still has an accuracy of 76.84%.

The experimental results for HE were realised with BFV in batched mode and Palisade library (version 1.1.6) on an Intel Core i7 with 4 cores at 3 GHz with 32 GB Ram on Ubuntu 18.04. The security level was set to 128 bits and the batch size used was of 8,192. Following this the overall 486,654 updates can be packed in only 60 ciphertexts (where each of the 8,192 slots contains one gradient update). Table 3.2 provides the overall homomorphic computation time

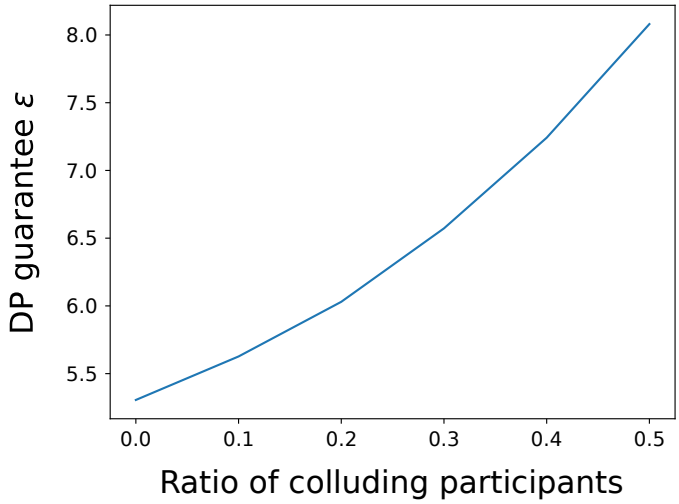

Figure 3.3: DP guarantee vs ratio of colluding participants ($\delta = 10^{-5}$)

Table 3.1: Influence of successive adaptations on accuracy.

	Accuracy
State of the art [122]	84.6%
Decrease the number of learning rounds T	83.58%
Increase M and K	81.04%
Assign same coefficients	80.21%
Clipping of the updates	79.26%
quantisation	79.03%
Modulo operation	79.07%
Adding random noise	**76.84%**

Table 3.2: Computation time (in seconds) of HE operations with a 26 bits modulus for the *full* 486654 weights model.

Number of participants K	1000
Context and key generation	$0,05698$
Encoding	$0,05704$
Encryption	$0,84642$
Evaluation	$26,22508$
Decryption	$0,29308$

for the full model for a 26 bits modulus and 1000 participants per round resulting in a (fairly practical) maximum of 26 seconds of homomorphic calculations (per FL round). Performing the full FL cycle (without communications) on a GPU-based HPC cluster takes around 20 hours (i.e., 12 minutes per FL round), resulting in a 3.6% computation time overhead imputable to HE calculations.

The choice of a 26 bits modulus is due to an empirical investigation. For 26 bits or more, the model trains correctly, with almost no impact of the modulo operation on the accuracy (see Table 3.1). Below 26 bits, the model does not learn at all. This sharp change of behaviour is due to the fact that the modulus exponentially depends on the number of bits and that the distribution of the quantised noised updates is actually very peaked - the ratio standard deviation over expectation is lower than 2.22×10^{-3}.

3.6 . Conclusion and perspectives

In this work, we addressed the problem of rigorously combining Differential Privacy and Homomorphic Encryption in order to strengthen the training data privacy of collaborative learning protocols. Starting from the popular FL framework we provided a number of confidentiality-oriented building blocks. Firstly, by having the clients add random noise on the information they send to the server, we made the learning mechanism differentially private from the point of view of any end-user of the model and that of the clients themselves. Secondly, following [122], we added a HE layer on the server side so that the server cannot see the updates coming from the clients, that may release sensitive information about the training data. Yet, the HE layer has a major impact on the random noise added to ensure DP, essentially because of the limited number of bits available for a reasonable computation time. However, we proved that this interference can be seamlessly dealt with in terms of privacy thanks to some adaptations among which a new carefully crafted quantisation operator that frees the method from the trade-off between privacy and computation time/communication.

We ran experiments on the FEMNIST dataset that illustrate the practicality of our approach in terms of accuracy, precision requirement and computation time, and we thoroughly analysed the cost of DP in accuracy compared to a non-DP baseline.

On the server side, the present work could be extended to cover more advanced threat models, making the learning process robust to a server who would, willingly or not, make mistakes in its computations. As argued in Section 3.4.7, this could be done using verifiable computing techniques, as in [122], in a quite straightforward further work thanks to the fact that the server is not in charge of adding the random noise necessary to DP. It should also be emphasised that our quantisation technique may also prove useful when combined with other cryptographic techniques for computing over encrypted data such as MPC and Functional Encryption which also have applications in Federated Learning. Of course, our quantisation operator also addresses the issue of communication overload even in a cryptography-free context.

Testing our approach on a larger, more cross-device-oriented dataset would be quite interesting to further estimate its scalability. Moreover, this could be advantageous from the privacy point of view since this would allow to increase M, the number of clients and thus having

simultaneously a large number of participants K and a low ratio $\frac{K}{M}$, conditions that will both improve the DP guarantees of the learning mechanism.

Another quantisation function or a more involved analysis that would not need to lower bound the random noise added to the updates would allow us to get rid of the argument of the imperfect sampling algorithms and to use our framework with other noise distributions, possibly unbounded, even in practice.

E - Sums of rounded Gaussian variables are not rounded Gaussian variables

In this appendix, we show that the sum of rounded Gaussian variables does not follow the distribution of a rounded Gaussian variable in general. We exhibit a counter-example for the sum of two rounded Gaussian variables.

Let $\lfloor \cdot \rceil$ be one of the two following operators: the deterministic rounding (rounding to the nearest integer) or the traditional stochastic rounding ($x \in \mathbb{R}$ is rounded to $\lfloor x \rfloor$ with probability $\lfloor x \rfloor + 1 - x$ and to $\lfloor x \rfloor + 1$ with probability $x - \lfloor x \rfloor$, where $\lfloor x \rfloor$ is the floor of x). We may apply this operator to a real or a real valued random variable.

Let X_1, X_2 be two independent random variables both following the normal law of mean $\frac{1}{2}$ and standard deviation $\sigma \in \mathbb{R}_+^*$.

When σ approaches 0, for all $i \in \{1, 2\}$, $\mathbb{P}(\lfloor X_i \rceil = 0)$ and $\mathbb{P}(\lfloor X_i \rceil = 1)$ both approach $\frac{1}{2}$. Let us then choose σ small enough, such that $\mathbb{P}(\lfloor X_i \rceil = 0) \geq \frac{1}{4}$ and $\mathbb{P}(\lfloor X_i \rceil = 1) \geq \frac{1}{4}$ for all $i \in \{1, 2\}$. As a consequence,

$$\begin{aligned} \mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 0) &\geq \mathbb{P}(\lfloor X_1 \rceil = 0 \wedge \lfloor X_2 \rceil = 0) \\ &= \mathbb{P}(\lfloor X_1 \rceil = 0)\mathbb{P}(\lfloor X_2 \rceil = 0) \\ &\geq \frac{1}{16}. \end{aligned} \tag{E.1}$$

Moreover,

$$\begin{aligned} \mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 1) &\geq \mathbb{P}(\lfloor X_1 \rceil = 0 \wedge \lfloor X_2 \rceil = 1) + \mathbb{P}(\lfloor X_1 \rceil = 1 \wedge \lfloor X_2 \rceil = 0) \\ &= \mathbb{P}(\lfloor X_1 \rceil = 0)\mathbb{P}(\lfloor X_2 \rceil = 1) + \mathbb{P}(\lfloor X_1 \rceil = 1)\mathbb{P}(\lfloor X_2 \rceil = 0) \\ &\geq \frac{1}{8}. \end{aligned} \tag{E.2}$$

Suppose that $\lfloor X_1 \rceil + \lfloor X_2 \rceil$ follows the distribution of a rounded Gaussian. We call Y the Gaussian variable such that $\lfloor X_1 \rceil + \lfloor X_2 \rceil$ follows the same distribution as $\lfloor Y \rceil$. Let σ' be the standard deviation of Y and μ' its mean.

X_1 and X_2 being independent and symmetric around $\frac{1}{2}$, $\lfloor X_1 \rceil + \lfloor X_2 \rceil$ is symmetric around 1 and so is $\lfloor Y \rceil$. In particular $\mathbb{P}(\lfloor Y \rceil = 0) = \mathbb{P}(\lfloor Y \rceil = 2)$. If $\mu' < 1$, $\mathbb{P}(\lfloor Y \rceil = 0) > \mathbb{P}(\lfloor Y \rceil = 2)$ and, similarly, if $\mu' > 1$, $\mathbb{P}(\lfloor Y \rceil = 0) < \mathbb{P}(\lfloor Y \rceil = 2)$. Thus $\mu' = 1$.

Then, when σ' approaches 0, $\mathbb{P}(\lfloor Y \rceil \neq 1)$ approaches 0. Since E.1 implies that $\mathbb{P}(\lfloor Y \rceil \neq 1) \geq \frac{1}{16}$, σ' is greater than a certain $s_1 \in \mathbb{R}_+^*$.

Moreover, when σ' approaches infinity, $\mathbb{P}(Y \in [a, b])$ approaches 0 for any $(a, b) \in \mathbb{R}^2$, $a < b$. Inequality E.2 implies that $\mathbb{P}(Y \in [0, 2]) \geq \frac{1}{8}$ so it also implies that σ' is lower than a certain $s_2 \in \mathbb{R}_+^*$.

$s_1 \leq \sigma' \leq s_2$ gives us a lower bound on $\mathbb{P}(Y \leq -1)$. Let $p \in]0; 1]$ be a strict lower bound for $\mathbb{P}(Y \leq -1)$. Hence $\mathbb{P}(\lfloor Y \rceil \leq -1) \geq \mathbb{P}(Y \leq -1) > p$.

The definition of p only comes from the fact that the Gaussian variable Y has mean 1 and satisfies $\mathbb{P}(\lfloor Y \rceil \neq 1) \geq \frac{1}{16}$ and $\mathbb{P}(Y \in [0,2]) \geq \frac{1}{8}$. As such, p is independent of σ as soon as σ is small enough to ensure $\mathbb{P}(\lfloor X_i \rceil = 0) \geq \frac{1}{4}$ and $\mathbb{P}(\lfloor X_i \rceil = 1) \geq \frac{1}{4}$ for all $i \in \{1,2\}$. Moreover, $p > 0$ and then $\frac{\sqrt{1-p}}{2} < \frac{1}{2}$. Hence, we can choose σ such that, besides $\mathbb{P}(\lfloor X_i \rceil = 0) \geq \frac{1}{4}$ and $\mathbb{P}(\lfloor X_i \rceil = 1) \geq \frac{1}{4}$, we also have $\mathbb{P}(\lfloor X_i \rceil = 0) \geq \frac{\sqrt{1-p}}{2}$ and $\mathbb{P}(\lfloor X_i \rceil = 1) \geq \frac{\sqrt{1-p}}{2}$, for all $i \in \{1,2\}$.

Then,

$$\begin{aligned}\mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 1) &\geq \mathbb{P}(\lfloor X_1 \rceil = 1 \wedge \lfloor X_2 \rceil = 0) + \mathbb{P}(\lfloor X_1 \rceil = 0 \wedge \lfloor X_2 \rceil = 1) \\ &= \mathbb{P}(\lfloor X_1 \rceil = 1)\mathbb{P}(\lfloor X_2 \rceil = 0) + \mathbb{P}(\lfloor X_1 \rceil = 0)\mathbb{P}(\lfloor X_2 \rceil = 1) \\ &\geq \frac{1-p}{2}. \end{aligned} \tag{E.3}$$

Besides,

$$\begin{aligned}\mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 0) &\geq \mathbb{P}(\lfloor X_1 \rceil = 0 \wedge \lfloor X_2 \rceil = 0) \\ &= \mathbb{P}(\lfloor X_1 \rceil = 0)\mathbb{P}(\lfloor X_2 \rceil = 0) \\ &\geq \frac{1-p}{4} \end{aligned} \tag{E.4}$$

and, by symmetry,

$$\mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 2) \geq \frac{1-p}{4}. \tag{E.5}$$

Inequalities E.3, E.4 and E.5 give

$$\begin{aligned}\mathbb{P}(\lfloor Y \rceil \leq -1) &\leq 1 - \mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 1) - \mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 0) - \mathbb{P}(\lfloor X_1 \rceil + \lfloor X_2 \rceil = 2) \\ &\leq p \end{aligned}$$

which gives a contradiction and shows that $\lfloor X_1 \rceil + \lfloor X_2 \rceil$ does not follow the distribution of a rounded Gaussian variable.

Conclusion

With the boom of machine learning and its ever-increasing greed for countless data from any kind of source, the privacy issues for artificial intelligence are subject to more and more tension and privacy requirements are often unnegotiable. In a nutshell, privacy-preserving machine learning is becoming a standard, enforced by constraining regulations around the world. Due to the variety of potential adversaries and attacks, many diverse techniques have been developed as countermeasures. Yet, the combination of these techniques in efficient and well integrated protocols still poses many challenges and constitutes an active field of research.

In our three contributions, we leveraged the complementarity of two of these techniques, differential privacy (DP) and homomorphic encryption (HE), to widen the scope of addressed threats and preserve the training data privacy against any actor of the learning protocol and the end-users. In the contributions of Chapters 1 and 3 of Part II, the combination of these two very different privacy tools appeared like a constraint. In SPEED (Chapter 1), the set of computations to realise in the encrypted domain is decided following a trade-off between revealing the less leaky output (argmax rather than histogram) and limiting the complexity of the computations in the homomorphic domain. In *Combining homomorphic encryption and differential privacy in federated learning* (Chapter 3), the quantisation induced by the encryption and the unsuitability of discrete noise mechanisms compelled us to propose a new quantisation operator. Thanks to this operator, we were able to combine additive HE - which only induces a computational time overhead of less than 5% - with distributed DP, letting the possibility to integrate verifiable computing techniques in the line of our other contribution *A secure federated learning framework using HE and verifiable computing*. On the contrary, SHIELD (Chapter 2 of Part II) makes DP and HE help each other and play together in the same direction: the approximate design of this operator makes it faster to compute in the homomorphic domain while turning it error-prone, thus ensuring DP. This opens a new paradigm of optimisation with an additional degree of freedom: the trade-off between accuracy, privacy and computational performance rather than a binary trade-off as privacy-accuracy in DP and security-performance in FHE.

Our works provide different solutions for privacy-preserving server-based collaborative learning. In the context of the recent privacy rules, like GDPR for example, a wide range of applications is possible. One of the most obvious application scenarios lies in the medical field: teachers or clients can be hospitals that own medical data from patients, do not want to jeopardise their privacy (think about HIPAA that specifically regulates privacy in healthcare) but aim at aggregating these data to train a global model that would, for instance, detect a certain disease from radiographic images. Defence is also an important application field where unwanted information leakage may result in serious consequences. Actually, DGA (Direction Générale de l'Armement)[1], interested by our SPEED solution, contacted us to design

[1]the French Government Defence agency responsible for the development, purchase and exportation of French weapon systems

a realistic application scenario against cyberattacks in the context of Opération i-Naval 2022 https://2022.i-naval.fr/. Another application field is cybersecurity. Indeed, many systems register signatures of attacks that they suffered in the past. If the register is discovered by an adversary, it could reveal the weaknesses of the system and make a future attack much more efficient. More generally, any learning framework that involves several parties that own sensitive data, either for personal or strategic reasons, is a potential application.

As suggested by our experimental results, the number of data owners involved can guide the choice of the framework: SPEED and SHIELD work well with a quite small number of teachers (a few hundreds) whereas our secure federated learning solution can deal with thousands of clients. Besides, SPEED approach is agnostic to the nature of the models and thus allows to work with large-scale models without additional cost in the homomorphic domain. In contrast, federated learning is specific to neural networks and limits the server computations to the aggregation and, in case of FedAvg, to a simple addition, reducing the homomorphic overhead to the minimum.

Precisely, among the perspectives of our line of research is the design of another aggregation operator for federated learning, which would be both quick to compute in the homomorphic domain and robust to Byzantine attacks i.e. attacks from clients that send incorrect updates, either random or adversarial. This perspective is challenging because the most common aggregation operators that are robust to Byzantine attacks (e.g. median, Krum, multi-Krum) make use of comparisons, which are costly operations in the homomorphic domain. We think that DP may also help to reach Byzantine robustness. In [86], the authors explain that DP is incompatible with the variance-to-norm (VN) condition that implies Byzantine resilience and needs to be relaxed to get along with DP. Nevertheless, since DP aims at obfuscating the contribution of one specific client (or even a group of clients, depending on the adjacency definition), this precisely could mitigate the malicious influence of a Byzantine adversary and, in some cases, DP and Byzantine robustness - apprehended from another point of view than the VN condition - might have aligned objectives.

Another extension to not only honest-but-curious adversaries could rely on verifiable computing. In [122], we already proposed a federated learning framework secured by verifiable computing and HE but it lacked DP. As stated in Chapter 3 of Part II, we could easily add a verifiable feature to our method thanks to the fact that the noise is generated in a distributed manner among the participants.

An interesting line of research is the use of the intrinsic noise induced by encryption to ensure DP guarantees. As noted in Chapter 2 of Part II, in some scenarios when we need both HE and DP, cryptologists may spend a lot of efforts to make the homomorphic layer accurate, which increases the computational complexity. Afterwards, some noise is added to ensure DP. One can feel that this approach is not optimal and the idea here is to let the HE operators be somehow inaccurate, enabling low computational complexity and, at the same time, providing DP guarantees. Since encryption is necessarily probabilistic, if we managed to characterise the distribution of the encryption noise, we could tune the cryptosystem parameters in order to get the required amount of noise for DP from encryption, which would also result in a lighter computational load than in the case of a very accurate operator. CKKS is a good candidate

for this approach since the ratio between the noise added at encryption and the message is more important than in most of the other cryptosystems. This follows the same philosophy as SHIELD but is closer to [171] because, rather than having a stochastic algorithm by design, we would get this stochastic behaviour from the noise induced by encryption.

Further work could also include wider reflections about DP, its limits and its possible relaxations. For instance, what is a reasonable ϵ remains unclear [64] and in many applications or even research works, the choice of ϵ is more or less arbitrary, sometimes leading to little to no privacy [23]. Even if the idea of the continuum on the privacy-utility axis on which lie both cryptography and DP, explained in Section 2.2.7 of Part I, gives a comparison point for the value of ϵ, the notion of security in cryptography is so conservative and so far from practical levels of "useful" privacy in DP that this comparison is of little help for the determination of a good epsilon value. Non-expert clients would highly benefit from automatic mechanisms that would allow them to control the privacy-utility trade-off more easily by, among other functionalities, suggesting reasonable values of ϵ and δ depending on the application, ensuring a maximum number of queries to a given database. A good starting point towards these mechanisms is the Epsilon Registry, a public collaborative document that gathers empirical knowledge about DP implementations, for which Dwork et al. call in [64].

Extensions of DP could be more adapted to the frameworks we worked on, especially SHIELD. Among them is the setting whereby it is not assumed that the adversary knows everything about all the individuals in the database except its victim, but that it may know less. As explained in [14, 20], the uncertainty on the individuals results in increased noise on the sensitive information, thus leading to better DP guarantees, or even privacy guarantees without noise addition. Using DP based on other metrics than the Hamming distance, as proposed in [46], would grasp a notion of privacy that goes beyond the pairs of adjacent databases and may be more suitable to certain settings. In particular, such an extended notion of DP could solve the paradox of counter-productive noise mentioned in the final remark of Appendix A.4 and especially in Appendix D.

Bibliography

[1] GDPR 2018 reform of eu data protection rules. https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf, 2018.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC*, pages 308–318, 2016.

[3] Gergely Ács and Claude Castelluccia. I have a dream!(differentially private smart metering). In *International Workshop on Information Hiding*, pages 118–132. Springer, 2011.

[4] Archita Agarwal, Maurice Herlihy, Seny Kamara, and Tarik Moataz. Encrypted databases for differential privacy. *Cryptology ePrint Archive*, 2018.

[5] Naman Agarwal, Peter Kairouz, and Ziyu Liu. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[6] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. *NeurIPS*, 31:7564–7575, 2018.

[7] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018.

[8] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169 – 203, 2015.

[9] Mário S Alvim, Miguel E Andrés, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the relation between differential privacy and quantitative information flow. In *International Colloquium on Automata, Languages, and Programming*, pages 60–76. Springer, 2011.

[10] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.

[11] Pascal Aubry, Sergiu Carpov, and Renaud Sirdey. Faster homomorphic encryption is not enough: improved heuristic for multiplicative depth minimization of boolean circuits. In *CT-RSA*, pages 345–363, 2019.

[12] Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. Openfhe: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915, 2022. https://eprint.iacr.org/2022/915.

[13] Haiyong Bao and Rongxing Lu. A new differentially private data aggregation with fault tolerance for smart grid communications. *IEEE Internet of Things Journal*, 2(3):248–258, 2015.

[14] Raef Bassily, Adam Groce, Jonathan Katz, and Adam Smith. Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 439–448. IEEE, 2013.

[15] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1046–1059, 2016.

[16] Johes Bater, Xi He, William Ehrich, Ashwin Machanavajjhala, and Jennie Rogers. Shrinkwrap: efficient sql query processing in differentially private data federations. *Proceedings of the VLDB Endowment*, 12(3), 2018.

[17] Brett K. Beaulieu-Jones, William Yuan, Samuel G. Finlayson, and Zhiwei Steven Wu. Privacy-preserving distributed deep learning for clinical data. *CoRR*, abs/1812.01484, 2018.

[18] Josh Cohen Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In *Conference on the theory and application of cryptographic techniques*, pages 251–260. Springer, 1986.

[19] Flavio Bergamaschi, Shai Halevi, Tzipora T Halevi, and Hamish Hunt. Homomorphic training of 30,000 logistic regression models. In *International Conference on Applied Cryptography and Network Security*, pages 592–611. Springer, 2019.

[20] Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta. Noiseless database privacy. In *Advances in Cryptology–ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings 17*, pages 215–232. Springer, 2011.

[21] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

[22] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld.

Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*, pages 441–459, 2017.

[23] Alberto Blanco-Justicia, David Sanchez, Josep Domingo-Ferrer, and Krishnamurty Muralidhar. A critical review on the use (and misuse) of differential privacy in machine learning. *arXiv preprint arXiv:2206.04621*, 2022.

[24] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

[25] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[26] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*, pages 253–273. Springer, 2011.

[27] Christina Boura, Nicolas Gama, and Mariya Georgieva. Chimera: a unified framework for b/fv, tfhe and heaan fully homomorphic encryption and predictions for deep learning. Cryptology ePrint Archive, Report 2018/758, 2018.

[28] Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. Fast homomorphic evaluation of deep discretized neural networks. In *Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38*, pages 483–512. Springer, 2018.

[29] George Edward Pelham Box and Mervin Edgar Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29:610–611, 1958.

[30] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 337–367. Springer, 2015.

[31] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption Without Bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, 2012.

[32] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory*, 6(3), jul 2014.

[33] S. Caldas, S.M. Karthik Duddu, P. Wu, T. Li, J. Konečný, H.B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. arXiv preprint 1812.01097, 2019.

[34] Clément Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

[35] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. *Journal of Cryptology*, 31:885–916, 2018.

[36] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[37] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.

[38] S. Carpov, N. Gama, M. Georgieva, and J. R. Troncoso-Pastoriza. Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption. Cryptology ePrint Archive, Report 2019/101, 2019. https://eprint.iacr.org/2019/101.

[39] Sergiu Carpov, Paul Dubrulle, and Renaud Sirdey. Armadillo: a compilation chain for privacy preserving applications. In *Proceedings of the 3rd International Workshop on Security in Cloud Computing*, pages 13–19, 2015.

[40] Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/, 1996.

[41] Hervé Chabanne, Amaury De Wargny, Jonathan Milgram, Constance Morel, and Emmanuel Prouff. Privacy-preserving classification on deep neural network. *Cryptology ePrint Archive*, 2017.

[42] Hervé Chabanne, Roch Lescuyer, Jonathan Milgram, Constance Morel, and Emmanuel Prouff. Recognition over encrypted faces. In *Mobile, Secure, and Programmable Networking: 4th International Conference, MSPN 2018, Paris, France, June 18-20, 2018, Revised Selected Papers 4*, pages 174–191. Springer, 2019.

[43] Olive Chakraborty and Martin Zuber. Efficient and accurate homomorphic comparisons. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC'22, pages 35–46. Association for Computing Machinery, 2022.

[44] T-H Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In *International Conference on Financial Cryptography and Data Security*, pages 200–214. Springer, 2012.

[45] Melissa Chase, Ran Gilad-Bachrach, Kim Laine, Kristin E Lauter, and Peter Rindal. Private collaborative neural network learning. *IACR Cryptology ePrint Archive*, 2017:762, 2017.

[46] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings 13*, pages 82–102. Springer, 2013.

[47] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

[48] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, pages 409–437. Springer, 2017.

[49] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pages 375–403. Springer, 2019.

[50] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT*, pages 3–33, 2016.

[51] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption library, August 2016. https://tfhe.github.io/tfhe/.

[52] Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized multi-client functional encryption for inner product. In *Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part II 24*, pages 703–732. Springer, 2018.

[53] Jérémy Chotard, Edouard Dufour-Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Dynamic decentralized functional encryption. In *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part I*, pages 747–775. Springer, 2020.

[54] Edwige Cyffers and Aurélien Bellet. Privacy amplification by decentralization. In *International Conference on Artificial Intelligence and Statistics*, pages 5334–5353. PMLR, 2022.

[55] Edwige Cyffers, Mathieu Even, Aurélien Bellet, and Laurent Massoulié. Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging. *arXiv preprint arXiv:2206.05091*, 2022.

[56] George Danezis, Cédric Fournet, Markulf Kohlweiss, and Santiago Zanella-Béguelin. Smart meter aggregation via secret-sharing. In *Proceedings of the first ACM workshop on Smart energy grid security*, pages 75–80, 2013.

[57] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1–5, 2013.

[58] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[59] Judith Wagner DeCew. *In pursuit of privacy: Law, ethics, and the rise of technology*. Cornell University Press, 1997.

[60] Damien Desfontaines and Balázs Pejó. Sok: differential privacies. *Proceedings on privacy enhancing technologies*, 2020(2):288–313, 2020.

[61] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

[62] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE, 2013.

[63] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[64] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019.

[65] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[66] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[67] Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

[68] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[69] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[70] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, pages 6246–6283. PMLR, 2022.

[71] Dario Fiore, Rosario Gennaro, and Valerio Pastro. Efficiently verifiable computation on encrypted data. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 844–855, 2014.

[72] Matthew Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 699–710, New York, NY, USA, 1992. Association for Computing Machinery.

[73] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC*, pages 1322–1333, 2015.

[74] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.

[75] Simson Garfinkel, John M Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Communications of the ACM*, 62(3):46–53, 2019.

[76] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[77] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[78] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *ICML*, pages 201–210, 2016.

[79] Shafi Goldwasser and Silvio Micali. Probabilistic encryption in jcss, 28 (2). *MATH MathSciNet*, pages 270–299, 1984.

[80] Slawomir Goryczka and Li Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE transactions on dependable and secure computing*, 14(5):463–477, 2015.

[81] Slawomir Goryczka, Li Xiong, and Vaidy Sunderam. Secure multiparty aggregation with differential privacy: A comparative study. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 155–163, 2013.

[82] Thore Graepel, Kristin Lauter, and Michael Naehrig. Ml confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology*, pages 1–21. Springer, 2012.

[83] Arnaud Grivet Sébert, Rafaël Pinot, Martin Zuber, Cedric Gouy-Pailler, and Renaud Sirdey. Speed: secure, private, and efficient deep learning. *Machine Learning*, 110(4):675–694, 2021.

[84] Arnaud Grivet Sébert, Renaud Sirdey, Oana Stan, and Cédric Gouy-Pailler. Protecting data from all parties: Combining fhe and dp in federated learning. *arXiv preprint arXiv:2205.04330*, 2022.

[85] Arnaud Grivet Sébert, Martin Zuber, Oana Stan, Renaud Sirdey, and Cédric Gouy-Pailler. When approximate design for fast homomorphic computation provides differential privacy guarantees. *arXiv preprint arXiv:2304.02959*, 2023.

[86] Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, Sebastien Rouault, and John Stephan. Combining differential privacy and byzantine resilience in distributed sgd. *arXiv preprint arXiv:2110.03991*, 2021.

[87] Awni Hannun, Chuan Guo, and Laurens van der Maaten. Measuring data leakage in machine-learning models with fisher information. In *Uncertainty in Artificial Intelligence*, pages 760–770. PMLR, 2021.

[88] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10):6532–6542, 2019.

[89] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[90] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*, 2017.

[91] E. Hesamifard, H. Takabi, and M. Ghasemi. Deep neural networks classification over encrypted data. In *ACM CODASPY*, page 97–108, 2019.

[92] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*, 2017.

[93] Seira Hidano, Takao Murakami, Shuichi Katsumata, Shinsaku Kiyomoto, and Goichiro Hanaoka. Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 115–11509. IEEE, 2017.

[94] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019(4):232–249, 2019.

[95] Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63, 2020.

[96] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[97] Ilia Iliashenko and Vincent Zucca. Faster homomorphic comparison operations for BGV and BFV. *Proceedings on Privacy Enhancing Technologies*, 2021(3):246–264, 2021. Publisher: De Gruyter Open.

[98] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Annual International Cryptology Conference*, pages 145–161. Springer, 2003.

[99] M. Izabachène, R. Sirdey, and M. Zuber. Practical fully homomorphic encryption for fully masked neural networks. In *Cryptology and Network Security - 18th International Conference, CANS 2019, Proceedings*, volume 11829 of *Lecture Notes in Computer Science*, pages 24–36. Springer, 2019.

[100] Jaehee Jang, Younho Lee, Andrey Kim, Byunggook Na, Donggeon Yhee, Byounghan Lee, Jung Hee Cheon, and Sungroh Yoon. Privacy-preserving deep sequential model with matrix homomorphic encryption. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 377–391, 2022.

[101] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. *NeurIPS*, 31:6343–6354, 2018.

[102] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[103] Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees. *arXiv preprint arXiv:1909.03577*, 2019.

[104] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1651–1669, 2018.

[105] P. Kairouz et al. Advances and open problems in federated learning. arXiv preprint 1912.04977, 2019.

[106] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pages 5201–5212. PMLR, 2021.

[107] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *The Journal of Machine Learning Research*, 17(1):492–542, 2016.

[108] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[109] Miran Kim, Yongsoo Song, Shuang Wang, Yuhou Xia, Xiaoqian Jiang, et al. Secure logistic regression based on homomorphic encryption: Design and evaluation. *JMIR medical informatics*, 6(2):e8805, 2018.

[110] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, third edition, 1997.

[111] Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 3358–3366. PMLR, 2021.

[112] S Kotz, TJ Kozubowski, and K Podgorski. The laplace distribution and generalizations, 2001.

[113] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[114] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010. *URL http://yann. lecun. com/exdb/mnist*, 7:23, 2010.

[115] Joon-Woo Lee, HyungChul Kang, Yongwoo Lee, Woosuk Choi, Jieun Eom, Maxim Deryabin, Eunsang Lee, Junghyun Lee, Donghoon Yoo, Young-Sik Kim, et al. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054, 2022.

[116] Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In *Annual International Cryptology Conference*, pages 560–589. Springer, 2022.

[117] Junyi Li and Heng Huang. Faster secure data mining via distributed homomorphic encryption. In *ACM SIGKDD*, pages 2706–2714, 2020.

[118] Shenghui Li, Edith C-H Ngai, and Thiemo Voigt. An experimental study of byzantine-robust aggregation schemes in federated learning. *IEEE Transactions on Big Data*, 2023.

[119] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[120] Qian Lou, Bo Feng, Geoffrey Charles Fox, and Lei Jiang. Glyph: Fast and accurately training deep neural networks on encrypted data. *Advances in Neural Information Processing Systems*, 33:9193–9202, 2020.

[121] A. Madi, O. Stan, and R. Sirdey. Computing neural networks with homomorphic encryption and verifiable computing. In *Proceedings of the 2nd Workshop on Cloud Security and Privacy*, number 12418 in LNCS, pages 295–317, 2020.

[122] Abbass Madi, Oana Stan, Aurélien Mayoue, Arnaud Grivet-Sébert, Cédric Gouy-Pailler, and Renaud Sirdey. A secure federated learning framework using homomorphic encryption and verifiable computing. In *2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS)*, pages 1–8. IEEE, 2021.

[123] Abbass Madi, Oana Stan, Renaud Sirdey, and Cédric Gouy-Pailler. Sectl: Secure and verifiable transfer learning-based inference. 2022.

[124] George Marsaglia and Wai Wan Tsang. The ziggurat method for generating random variables. *Journal of statistical software*, 5:1–7, 2000.

[125] AM Mathai. On noncentral generalized laplacianness of quadratic forms in normal variables. *Journal of multivariate analysis*, 45(2):239–246, 1993.

[126] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[127] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private language models without losing accuracy. *arXiv preprint arXiv:1710.06963*, 2017.

[128] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[129] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[130] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[131] Darakhshan J Mir. Information-theoretic foundations of differential privacy. In *Foundations and Practice of Security: 5th International Symposium, FPS 2012, Montreal, QC, Canada, October 25-26, 2012, Revised Selected Papers 5*, pages 374–381. Springer, 2013.

[132] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[133] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In *Advances in Cryptology-CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 126–142. Springer, 2009.

[134] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. Smpai: Secure multi-party computation for federated learning. In *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

[135] Arjun Narayan and Andreas Haeberlen. Djoin: Differentially private join queries over distributed databases. In *Presented as part of the 10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*, pages 149–162, 2012.

[136] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[137] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[138] Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA l. Rev.*, 57:1701, 2009.

[139] Brooks Paige, James Bell, Aurélien Bellet, Adrià Gascón, and Daphne Ezer. Reconstructing genotypes in private genomic databases from genetic risk scores. *Journal of Computational Biology*, 28(5):435–451, 2021.

[140] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE, 2020.

[141] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2016.

[142] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

[143] European Parliament and European Council. Regulation (eu) 2016/679 of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec. Technical report, European Parliament and European Council, 2016.

[144] Jestine Paul, Meenatchi Sundaram Muthu Selva Annamalai, William Ming, Ahmad Al Badawi, Bharadwaj Veeravalli, and Khin Mi Mi Aung. Privacy-preserving collective learning with homomorphic encryption. *IEEE Access*, 9:132084–132096, 2021.

[145] A. Pedrouzo-Ulloa, Aymen Boudguiga, Olive Chakraborty, Renaud Sirdey, Oana Stan, and Martin Zuber. Practical multi-key homomorphic encryption for more flexible and efficient secure federated aggregation. Technical Report 2022/1674, IACR ePrint, 2022.

[146] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[147] NhatHai Phan, Xintao Wu, and Dejing Dou. Preserving differential privacy in convolutional deep belief networks. *Machine learning*, 106(9-10):1681–1704, 2017.

[148] NhatHai Phan, Xintao Wu, Han Hu, and Dejing Dou. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In *2017 IEEE international conference on data mining (ICDM)*, pages 385–394. IEEE, 2017.

[149] Le Trieu Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-preserving deep learning via additively homomorphic encryption. Cryptology ePrint Archive, Report 2017/715, 2017. https://eprint.iacr.org/2017/715.

[150] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 735–746, 2010.

[151] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902. IEEE, 2015.

[152] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646*, 2020.

[153] Théo Ryffel, David Pointcheval, and Francis Bach. Ariann: Low-interaction privacy-preserving deep learning via function secret sharing. *arXiv preprint arXiv:2006.04593*, 2020.

[154] Théo Ryffel, Edouard Dufour Sans, Romain Gay, Francis Bach, and David Pointcheval. Partially encrypted machine learning using functional encryption. *arXiv preprint arXiv:1905.10214*, 2019.

[155] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

[156] César Sabater, Aurélien Bellet, and Jan Ramon. Distributed differentially private averaging with improved utility and robustness to malicious parties. *arXiv preprint arXiv:2006.07218*, 2020.

[157] César Sabater, Aurélien Bellet, and Jan Ramon. An accurate, scalable and verifiable protocol for federated differentially private averaging. *Machine Learning*, pages 1–45, 2022.

[158] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*, pages 5558–5567, 2019.

[159] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[160] A. Sanyal, M. Kusner, A. Gascón, and V. Kanade. TAPAS: Tricks to accelerate (encrypted) prediction as a service. In *ICML*, 06 2018.

[161] Sinem Sav, Apostolos Pyrgelis, Juan R Troncoso-Pastoriza, David Froelicher, Jean-Philippe Bossuat, Joao Sa Sousa, and Jean-Pierre Hubaux. POSEIDON: Privacy-preserving federated neural network learning. *arXiv preprint arXiv:2009.00349*, 2020.

[162] Ferdinand David Schoeman. *Philosophical dimensions of privacy: An anthology*. Cambridge University Press, 1984.

[163] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[164] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.

[165] Elaine Shi, TH Hubert Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Proc. NDSS*, volume 2, pages 1–17. Citeseer, 2011.

[166] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *ACM SIGSAC*, pages 1310–1321, 2015.

[167] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE SP*, pages 3–18. IEEE, 2017.

[168] Renaud Sirdey, Arnaud Grivet Sébert, and Cédric Gouy-Pailler. [cahier technique] cryptographie homomorphe : l'art de partager sans divulguer. *Industrie et technologies*, 2022.

[169] N. P. Smart and F. Vercauteren. Fully homomorphic simd operations. Cryptology ePrint Archive, Paper 2011/133, 2011. https://eprint.iacr.org/2011/133.

[170] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390, 2020.

[171] Timothy Stevens, Christian Skalka, Christelle Vincent, John Ring, Samuel Clark, and Joseph Near. Efficient differentially private secure aggregation for federated learning via hardness of learning with errors. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1379–1395, 2022.

[172] Daphné Trama, Pierre-Emmanuel Clet, Aymen Boudguiga, and Renaud Sirdey. Building blocks for lstm homomorphic evaluation with tfhe. *Proceedings of the International Symposium on Cyber Security, Cryptology and Machine Learning (CSCML), 2023 (to appear)*, 2023.

[173] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016.

[174] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11, 2019.

[175] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 14(6):2073–2089, 2019.

[176] Jonathan Ullman. Tight lower bounds for locally differentially private selection. *arXiv preprint arXiv:1802.02638*, 2018.

[177] Ayse Unsal and Melek Onen. Information-theoretic approaches to differential privacy. *arXiv preprint arXiv:2203.11804*, 2022.

[178] Jelle Van Den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, pages 137–152, 2015.

[179] Sameer Wagh, Paul Cuff, and Prateek Mittal. Differentially private oblivious ram. *Proceedings on Privacy Enhancing Technologies*, 2018(4):64–84, 2018.

[180] Sameer Wagh, Xi He, Ashwin Machanavajjhala, and Prateek Mittal. Dp-cryptography: marrying differential privacy and cryptography in emerging applications. *Communications of the ACM*, 64(2):84–93, 2021.

[181] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 36–52. IEEE, 2018.

[182] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[183] Samuel Warren and Louis Brandeis. The right to privacy. In *Killing the Messenger*, pages 1–21. Columbia University Press, 1989.

[184] Alan F Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1):166, 1968.

[185] Jongho Won, Chris YT Ma, David KY Yau, and Nageswara SV Rao. Proactive fault-tolerant aggregation protocol for privacy-assured smart metering. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 2804–2812. IEEE, 2014.

[186] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370. IEEE, 2016.

[187] Depeng Xu, Shuhan Yuan, Xintao Wu, and HaiNhat Phan. Dpne: Differentially private network embedding. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II 22*, pages 235–246. Springer, 2018.

[188] Mengjia Yan, Christopher W. Fletcher, and Josep Torrellas. Cache telepathy: Leveraging shared resource attacks to learn DNN architectures. *CoRR*, abs/1808.04761, 2018.

[189] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM TIST*, 10(2):1–19, 2019.

[190] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[191] Oualid Zari, Javier Parra-Arnau, Ayşe Ünsal, Thorsten Strufe, and Melek Önen. Membership inference attack against principal component analysis. In *Privacy in Statistical Databases: International Conference, PSD 2022, Paris, France, September 21–23, 2022, Proceedings*, pages 269–282. Springer, 2022.

[192] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, pages 493–506, 2020.

[193] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.

[194] Wenting Zheng, Raluca Ada Popa, Joseph E Gonzalez, and Ion Stoica. Helen: Maliciously secure coopetitive learning for linear models. In *2019 IEEE SP)*, pages 724–738. IEEE, 2019.

[195] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[196] Wennan Zhu, Peter Kairouz, Brendan McMahan, Haicheng Sun, and Wei Li. Federated heavy hitters discovery with differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 3837–3847. PMLR, 2020.

[197] Martin Zuber, Sergiu Carpov, and Renaud Sirdey. Towards real-time hidden speaker recognition by means of fully homomorphic encryption. Cryptology ePrint Archive, Report 2019/976, 2019.