Machine Learning (2021) 110:675–694
https://doi.org/10.1007/s10994-021-05970-3

SPEED: secure, PrivatE, and efficient deep learning

Arnaud Grivet Sébert[1] · **Rafaël Pinot**[1,2] · **Martin Zuber**[1] · **Cédric Gouy-Pailler**[1] · **Renaud Sirdey**[1]

Received: 22 November 2020 / Revised: 28 January 2021 / Accepted: 4 March 2021 /
Published online: 23 March 2021

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.

Keywords Data protection · Collaborative learning · Distributed differential privacy · Homomorphic encryption

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Arnaud Grivet Sébert
arnaud.grivetsebert@cea.fr

[1] Université Paris-Saclay, CEA, List, 91120 Palaiseau, France

[2] Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France

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 (Parliament and Council 2016). 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 (Shokri et al. 2017) 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 (Wu et al. 2016) 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 (Tramèr et al. 2016; Yan et al. 2018; Wang and Gong 2018) 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 Fig. 1 and formally detailed in Sect. 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

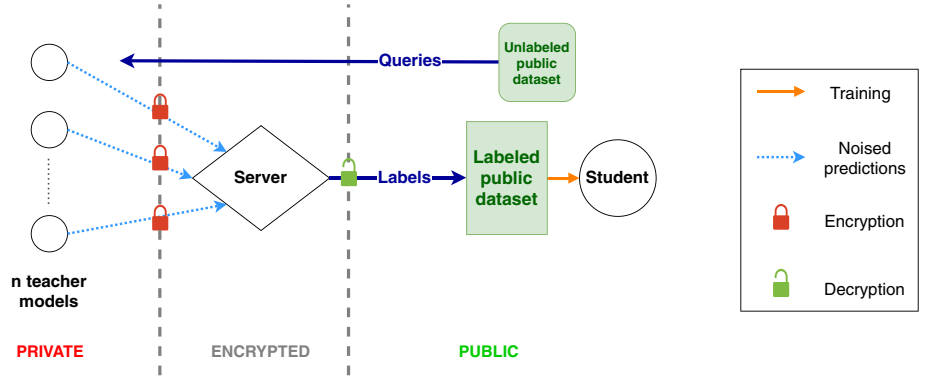

Fig. 1 SPEED—teacher models send to the aggregation server their encrypted noisy answers to the student's queries. The server homomorphically performs the aggregation in the encrypted domain and sends the result to the student model which decrypts it and uses it for training

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 Sect. 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 (Ishai et al. 2003; Bonawitz et al. 2016; Graepel et al. 2012).

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 2 relates our work to the literature. In Sect. 3, we give some technical background on differential privacy and homomorphic encryption. We describe our SPEED framework in Sect. 4 and analyse its differential privacy guarantees in Sect. 5. Section 6 presents our experimental results - SPEED achieves state-of-the-art accuracy and privacy with a mild computational overhead w.r.t previous works. Section 7 concludes the paper and states some open questions for further works.

2 Related work

Differential privacy (DP) Recent works considered to use differential privacy in collaborative settings close to the one we consider (Beaulieu-Jones et al. 2018; Bhowmick et al. 2018; Geyer et al. 2017; Chase et al. 2017; Papernot et al. 2016, 2018). Among them, the most efficient technique in terms of accuracy and privacy guarantees is Private Aggregation of Teacher Ensembles (PATE) first presented in Papernot et al. (2016) and refined in Papernot et al. (2018). 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 (Graepel et al. 2012; Hesamifard et al. 2017; Lou et al. 2020) and inference (Gilad-Bachrach et al. 2016; Juvekar et al. 2018) 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(McMahan et al. 2016) 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 (Geyer et al. 2017; Shokri and Shmatikov 2015), cryptographic primitives (Bonawitz et al. 2016, 2017; Ryffel et al. 2020) or both (Chase et al. 2017; Sabater et al. 2020; Ryffel et al. 2018). 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 (Kasiviswanathan et al. 2011; Duchi et al. 2013; Kairouz et al. 2016). In practice it often results in applying too much noise, and maintaining utility can be difficult (Ullman 2018; Kasiviswanathan et al. 2011) 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 Shi et al. 2011) and encryption schemes (Rastogi and Nath 2010; Ács and Castelluccia 2011; Goryczka and Xiong 2015; Shi et al. 2011) 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 (Bao and Lu 2015) while other works assume that the users themselves adapt the noise they generate to the possible failures (Chan et al. 2012). 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 Sect. 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 (Bonawitz et al. 2016, 2017) or secret-sharing (Danezis et al. 2013) 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 (Bonawitz et al. 2016, 2017). 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).

3 Preliminaries

3.1 Differential privacy

Differential privacy (Dwork et al. 2006) is a gold standard concept in privacy preserving data analysis. It provides a guarantee that under a reasonable privacy budget (ϵ, δ), two adjacent databases produce statistically indistinguishable results. In this section, two databases d and d' are said adjacent if they differ by at most one example.

Definition 1 A randomised mechanism $\mathcal{A}$ with output range $\mathcal{R}$ satisfies (ϵ, δ) *-differential privacy* if for any two adjacent databases d, d' and for any subset of outputs $S \subset \mathcal{R}$ one has

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

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

Definition 2 Let $K \in \mathbb{N}^*$, and let $\mathcal{X}$ be a set that can be partitioned into K subsets $\mathcal{X}_1$, ..., $\mathcal{X}_K$. The mechanism that, given a database d of elements of $\mathcal{X}$, reports $\operatorname{argmax}_{k\in[K]}\left[n_k + Y_k\right]$, where $[K] := \{1, \dots, K\}$, $n_k := |d \cap \mathcal{X}_k|$ and Y_k is a Laplace noise with mean 0 and scale $\frac{1}{\gamma}$, $\gamma \in \mathbb{R}_+^*$, is called *report noisy max*.

Theorem 1 (Dwork et al. 2014) *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 3 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 Kotz et al. (2001) claims that the Laplace distribution is infinitely divisible,[1] enabling to distribute its generation among an arbitrary number of agents.

Proposition 1 (Kotz et al. 2001) *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] 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.

Definition 4 Let $\mathcal{A}$ be a randomised mechanism with output range $\mathcal{R}$ and d, d' a pair of adjacent databases. Let aux denote an auxiliary input. For any $o \in \mathcal{R}$, the *privacy loss* at o is defined as

$$c(o;\mathcal{A}, \text{aux}, d, d') := \log\left(\frac{\mathbb{P}[\mathcal{A}(\text{aux}, d) = o]}{\mathbb{P}[\mathcal{A}(\text{aux}, d') = o]}\right).$$

We define the *privacy loss random variable* $C(\mathcal{A}, \text{aux}, d, d')$ as

$$C(\mathcal{A}, \text{aux}, d, d') := c(\mathcal{A}(d);\mathcal{A}, \text{aux}, d, d')$$

i.e. the random variable defined by evaluating the privacy loss at an outcome sampled from $\mathcal{A}(d)$.

In order to determine the privacy loss of our protocol, we use a traditional two-fold approach. First of all, we determine the privacy loss per query and, in a second step, we compose the privacy losses of each query to get the overall loss. The classical composition theorem (see e.g. Dwork et al. 2014) states that the guarantees ϵ of sequential queries add up. Nevertheless, training a deep neural network, even with a collaborative framework as presented in this paper, requires a large amount of calls to the databases, precluding the use of this classical composition. Therefore, to obtain reasonable DP guarantees, we need to keep track of the privacy loss with a more refined tool, namely the moments accountant (Abadiet al. 2016) that we introduce here, deferring the details of the method in Section A.1 of the appendix.

Definition 5 With the same notations as above, the *moments accountant* is defined for any $l \in \mathbb{R}_+^*$ as

$$\alpha_{\mathcal{A}}(l) := \max_{\text{aux},d,d'} \alpha_{\mathcal{A}}(l;\text{aux}, d, d')$$

where the maximum is taken over any auxiliary input aux and any pair of adjacent databases (d, d') and $\alpha_{\mathcal{A}}(l;\text{aux}, d, d') := \log\left(\mathbb{E}\left[\exp(lC(\mathcal{A}, \text{aux}, d, d'))\right]\right)$ is the moment generating function of the privacy loss random variable.

3.2 Homomorphic encryption

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). A homomorphic encryption 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.

Homomorphic encryption 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 bigger than the cleartexts set Λ. Any (decent) homomorphic encryption scheme possesses the semantic security property meaning that, given $\mathsf{Enc}(m)$ and polynomially many pairs $(m_i, \mathsf{Enc}(m_i))$ it is hard[2] to gain any information on m with a significant

[2] "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

advantage over guessing. Most importantly, a homomorphic encryption scheme offers two other operators $\oplus$ and $\otimes$ where

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

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 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[3]. *In practice*, though, the $\oplus$ and $\otimes$ are much more computationally costly than their clear domain counterparts which has led to the development of several approaches to HE schemes design each with their pros and cons.

Somewhat HE (SHE). Somewhat homomorphic encryption schemes, such as BGV (Brakerski et al. 2012) or BFV (Fan and Vercauteren 2012), 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 Aubry et al. 2019). 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.,

$$\mathsf{Enc}(m_1, \ldots, m_\kappa) \oplus \mathsf{Enc}(m'_1, \ldots, m'_\kappa) = \mathsf{Enc}(m_1 + m'_1, \ldots, m_\kappa + m'_\kappa) \tag{1}$$

(and similarly so for $\otimes$). Typically, several hundreds such slots are available which often allows to significantly speed up encrypted-domain calculations.

Fully HE (FHE). Fully homomorphic encryption 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 (Chillotti et al. 2016), offers practical performances. In this cryptosystem, $\oplus$ and $\otimes$ have the same constant cost. On the downside, TFHE offers no batching capabilities. 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 (Boura et al. 2018; Lou et al. 2020) to perform each part of calculation with the most appropriate scheme (see e.g. Zuber et al. 2020).

4 SPEED: secure, private, and efficient deep learning

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

Footnote 2 (continued)
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.

[3] Polynomial in λ.

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 $[K] = \{1, \ldots, K\}$. To do so while keeping the process private, we follow the setting illustrated by Fig. 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 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 Sect. 4.2) and encrypts the noisy label before sending it to the aggregation server.
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 Sect. 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 one hand, differential privacy protects the sensitive data from attacks against the student model. Indeed, some model inversion attacks (Wu et al. 2016) 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.[4] 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.

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 Sect. 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 the aggregator to add its own noise is even

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

Table 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

more harmful for the privacy (and of course, the accuracy) than not using noise at all. 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 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 $k \in [K]\backslash\{k_0, k_1\}$ 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 $\chi = \chi_0$ [5]. 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 1) 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 1).

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

- 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

Table 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, we focus on the case where 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 tradeoff between flexibility and security.

[5] 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.

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 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.

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 Zuber et al. (2020). They provide a fully non-interactive homomorphic argmax computation scheme based on the TFHE encryption. We implemented and parametrised their scheme to fit the specific training problems presented in Sect. 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 boostrap 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 Sect. 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-flavored) 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.

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 $i \in [n] \backslash \{i_0\}$, $f_i = f'_i$. This definition of adjacency is quite conservative and is strictly larger than the definition of adjacency from Papernot et al. (2016) (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 As we have decided not to trust the aggregation server to generate the noise necessary to the privacy guarantees, we may also assume that a subset of teachers might be malicious and collude by communicating their generated noise, which gives the same DP guarantees from the point of view of a colluding teacher as if they would have not generated any noise and, 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 2 *Let us define* $I : v \in \mathbb{R}_+^* \mapsto \int_0^{+\infty} (t+v)^{\tau-1} t^{\tau-1} e^{-2t} dt$ *and* $g : t \in \mathbb{R} \mapsto \frac{\int_{\gamma t}^{+\infty} e^{-v} I(v) dv}{\int_{\gamma(t+2)}^{+\infty} e^{-v} I(v) dv}$.

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 Dwork et al. (2014), 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 $t \in [-1;0]$, $g(t) \leq 1 + 2\frac{\int_0^{\gamma} e^{-\nu} I(v)dv}{\int_{2\gamma}^{+\infty} e^{-\nu} I(v)dv}$. On the other hand, we prove that, if besides $\tau > \frac{1}{2}$, then g is concave on $[\text{argmax}(g);0]$ and thus, for all $t \in [-1;0]$, $g(t) \leq g(0) - g'(0)$ (note that g is not differentiable in 0 if $\tau \leq \frac{1}{2}$). □

Denoting S the subset of teachers who are honest (i.e. do not collude), 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 $\tau = \frac{n-|S|}{n}$
- any entity who has access to the student model but is not a teacher, taking $\tau = 1$

Note that we can also use Theorem 2 in the hypothesis whereby the colluding teachers publish their noise (to the whole world), adapting τ in consequence.[6] For $\tau = 1$, the privacy guarantee is given by $\lim_{\tau \to 1} \epsilon'$ which, as shown by Proposition 2, is the classical bound of the *report noisy max* with a centralised Laplace noise.

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

Furthermore, Proposition 3 shows that, naturally, the privacy cost tends to be null when the noise becomes infinitely large (γ approaches 0).

Proposition 3 *For all* $\tau \in (0, 1)$, $\lim_{\gamma \to 0} \left[\log \left(1 + 2\frac{\int_0^{\frac{\gamma}{2}} e^{-\nu} I(v)dv}{\int_{\gamma}^{+\infty} e^{-\nu} I(v)dv} \right) \right] = 0$.

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

Proposition 4 *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 $\tau \in (0;\frac{1}{2}]$,

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

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

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 $k \in [K] \backslash \{k^*\}$, 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 $k \in [K] \backslash \{k^*\}$,

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

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 2 and Proposition 4 serve as building blocks to which we apply the following theorem from Papernot et al. (2016).

Theorem 3 (Papernot et al. 2016) *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*

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

As in Papernot et al. (2016), Theorem 3 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 Sect. 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 perturbates 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.

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

Table 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 ($\pm$2.27)	–
Trusted	1.41	95.95 ($\pm$2.97)	–
$\tau = 1$	1.41	95.91 ($\pm$2.57)	6.5min
$\tau = 0.9$	1.66	96.02 ($\pm$2.92)	
$\tau = 0.7$	2.37	96.06 ($\pm$2.61)	

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 Sect. 4.3. Without parallelizing, a single argmax query requires just under 4 s 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 min and 1000 queries 65 min. 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 Sect. 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 LeCun (1998), 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 LeCun (1998) and SVHN Netzer et al. (2011) 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.

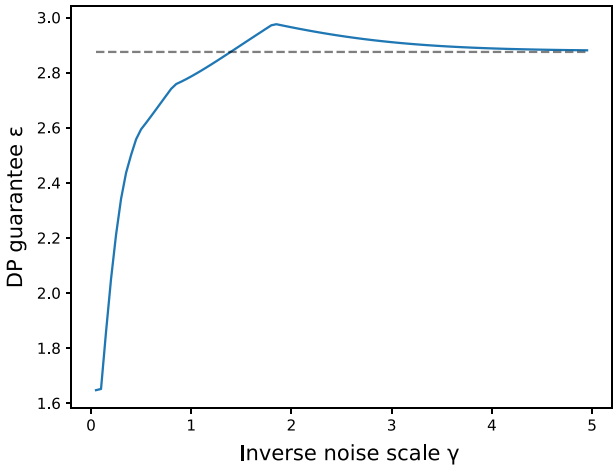

Fig. 2 Differential privacy guarantees for MNIST as a function of γ, with $\tau = 0.9$

– *Student model* Following the idea from Papernot et al. (2016), 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 Salimans et al. (2016). 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 10, 000 ones. We used 16, 032 examples to test the student model accuracy.

Performances on MNIST Table 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 Papernot et al. (2016) 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 Papernot et al. (2016) (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 tradeoff 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, 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

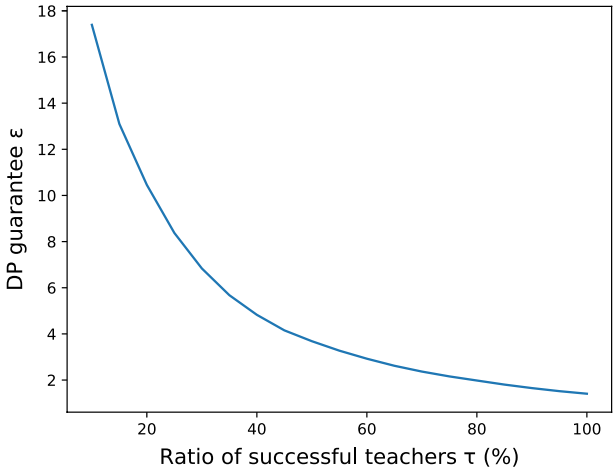

Fig. 3 Differential privacy guarantees for MNIST as a function of τ, with $\gamma = 0.1$

Table 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	

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%.

Figure 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 counterintuitive 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 Fig. 2.

Figure 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 3 presents our experimental results on SVHN dataset.[7] The variance on the accuracy is much smaller than for MNIST dataset because the test set is constituted of 16, 032 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

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

high amount of queries (500) necessary to obtain a good accuracy because the learning task is more complex.

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, Goryczka et al. 2013). 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. Fiore et al. 2014) 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.

Funding The experiments were performed using HPC resources of FactoryIA partially funded by Ile-de-France French region project SESAME 2017.

Declarations

Conflict of interest Not applicable.

Availability of data and material The MNIST (LeCun 1998) and SVHN (Netzer et al. 2011) datasets can be found respectively at http://yann.lecun.com/exdb/mnist/ and http://ufldl.stanford.edu/housenumbers/.

Code availability The source codes used to run the experiments and compute the DP guarantees can be accessed on https://github.com/Arnaud-GS/SPEED.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318

Ács, G., & Castelluccia, C. (2011). I have a dream!(differentially private smart metering). In: International Workshop on Information Hiding, pp. 118–132. Springer

Aubry, P., Carpov, S., & Sirdey, R. (2019). Faster homomorphic encryption is not enough: Improved heuristic for multiplicative depth minimization of boolean circuits. In: CT-RSA, pp. 345–363

Bao, H., & Lu, R. (2015). A new differentially private data aggregation with fault tolerance for smart grid communications. *IEEE Internet of Things Journal*, *2*(3), 248–258.
Beaulieu-Jones, B.K., Yuan, W., Finlayson, S.G., & Wu, Z.S. (2018). Privacy-preserving distributed deep learning for clinical data. CoRR **abs/1812.01484**
Bhowmick, A., Duchi, J., Freudiger, J., Kapoor, G., & Rogers, R. (2018). Protection against reconstruction and its applications in private federated learning. arXiv:1812.00984
Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2016). Practical secure aggregation for federated learning on user-held data. arXiv:1611.04482
Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191
Boura, C., Gama, N., & Georgieva, M. (2018). Chimera: A unified framework for b/fv, tfhe and heaan fully homomorphic encryption and predictions for deep learning. Cryptology ePrint Archive, Report 2018/758
Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2012). (Leveled) Fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12, pp. 309–325
Chan, T.H.H., Shi, E., & Song, D. (2012). Privacy-preserving stream aggregation with fault tolerance. In: International Conference on Financial Cryptography and Data Security, pp. 200–214. Springer
Chase, M., Gilad-Bachrach, R., Laine, K., Lauter, K. E., & Rindal, P. (2017). Private collaborative neural network learning. IACR Cryptology ePrint Archive, 2017, 762.
Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2016). Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: ASIACRYPT, pp. 3–33
Danezis, G., Fournet, C., Kohlweiss, M., & Zanella-Béguelin, S. (2013). Smart meter aggregation via secret-sharing. In: Proceedings of the First ACM Workshop on Smart Energy Grid Security, pp. 75–80
Duchi, J.C., Jordan, M.I., & Wainwright, M. J. (2013). Local privacy and statistical minimax rates. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pp. 429–438. IEEE
Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., & Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 486–503. Springer
Dwork, C., & Roth, A., et al. (2014). The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407
Fan, J., & Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, 2012, 144.
Fiore, D., Gennaro, R., & Pastro, V. (2014). Efficiently verifiable computation on encrypted data. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 844–855
Geyer, R.C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client level perspective. arXiv:1712.07557
Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning, pp. 201–210
Goryczka, S., & Xiong, L. (2015). A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, *14*(5), 463–477.
Goryczka, S., Xiong, L., & Sunderam, V. (2013). Secure multiparty aggregation with differential privacy: A comparative study. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops, pp. 155–163
Graepel, T., Lauter, K., & Naehrig, M. (2012). Ml confidential: Machine learning on encrypted data. In: International Conference on Information Security and Cryptology, pp. 1–21. Springer
Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. arXiv:1711.05189
Ishai, Y., Kilian, J., Nissim, K., & Petrank, E. (2003). Extending oblivious transfers efficiently. In: Annual International Cryptology Conference, pp. 145–161. Springer
Juvekar, C., Vaikuntanathan, V., & Chandrakasan, A. (2018). {GAZELLE}: A low latency framework for secure neural network inference. In: 27th {USENIX} Security Symposium ({USENIX} Security 18), pp. 1651–1669
Kairouz, P., Oh, S., & Viswanath, P. (2016). Extremal mechanisms for local differential privacy. *The Journal of Machine Learning Research*, *17*(1), 492–542.
Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., & Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, *40*(3), 793–826.

Kotz, S., Kozubowski, T., & Podgorski, K. (2012). *The Laplace distribution and generalizations: A revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media.

LeCun, Y. (1998). The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/

Lou, Q., Feng, B., Fox, G. C., & Jiang, L. (2020). Glyph: Fast and accurately training deep neural networks on encrypted data. *Advances in Neural Information Processing Systems*, *33*.

McMahan, H. B., Moore, E., Ramage, D., & Agüera y Arcas, B. (2016). Federated learning of deep networks using model averaging. arXiv:1602.05629.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., & Talwar, K. (2017). Semi-supervised knowledge transfer for deep learning from private training data. In *5th international conference on learning representations*.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., & Erlingsson, U. (2018). Scalable private learning with pate. In *6th international conference on learning representations*.

Parliament, E., & Council, E. (2016). 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. *European Parliament and European Council: Tech. rep*.

Rastogi, V., & Nath, S. (2010). Differentially private aggregation of distributed time-series with transformation and encryption. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 735–746).

Ryffel, T., Pointcheval, D., & Bach, F. (2020). Ariann: Low-interaction privacy-preserving deep learning via function secret sharing. arXiv:2006.04593

Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., & Passerat-Palmbach, J. (2018). A generic framework for privacy preserving deep learning. arXiv:1811.04017

Sabater, C., Bellet, A., & Ramon, J. (2020). Distributed differentially private averaging with improved utility and robustness to malicious parties. arXiv:2006.07218

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. arXiv:1606.03498

Shi, E., Chan, T.H., Rieffel, E., Chow, R., & Song, D. (2011). Privacy-preserving aggregation of time-series data. In: Proc. NDSS, vol. 2, pp. 1–17. Citeseer

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18. IEEE

Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., & Ristenpart, T. (2016). Stealing machine learning models via prediction apis. In: 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 601–618

Ullman, J. (2018). Tight lower bounds for locally differentially private selection. arXiv:1802.02638

Wang, B., & Gong, N. Z. (2018). Stealing hyperparameters in machine learning. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 36–52. IEEE

Wu, X., Fredrikson, M., Jha, S., & Naughton, J. F. (2016). A methodology for formalizing model-inversion attacks. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF), pp. 355–370. IEEE

Yan, M., Fletcher, C.W., & Torrellas, J. (2018). Cache telepathy: Leveraging shared resource attacks to learn DNN architectures. CoRR **abs/1808.04761**

Zuber, M., Carpov, S., & Sirdey, R. (2020). Towards real-time hidden speaker recognition by means of fully homomorphic encryption. In: International Conference on Information and Communications Security, pp. 403–421. Springer