Towards privacy-preserving and fairness-aware federated learning framework

Adda-Akram Bendoukha, Didem Demirag, Nesrine Kaaniche, Aymen Boudguiga, Renaud Sirdey, Sébastien Gambs

Towards Privacy-preserving and Fairness-aware Federated Learning Framework

Adda-Akram Bendoukha
Samovar, Télécom SudParis, Institut Polytechnique de Paris, France
adda-akram.bendoukha@telecom-sudparis.eu

Didem Demirag
Université du Québec à Montréal (UQAM), Canada
demirag.didem@uqam.ca

Nesrine Kaaniche
Samovar, Télécom SudParis, Institut Polytechnique de Paris, France
kaaniche.nesrine@telecom-sudparis.eu

Aymen Boudguiga
CEA List, Université Paris-Saclay, France
aymen.boudguiga@cea.fr

Renaud Sirdey
CEA List, Université Paris-Saclay, France
renaud.sirdey@cea.fr

Sébastien Gambs
Université du Québec à Montréal (UQAM), Canada
gambs.sebastien@uqam.ca

Abstract

Federated Learning (FL) enables the distributed training of a model across multiple data owners under the orchestration of a central server responsible for aggregating the models generated by the different clients. However, the original approach of FL has significant shortcomings related to privacy and fairness requirements. Specifically, the observation of the model updates may lead to privacy issues, such as membership inference attacks, while the use of imbalanced local datasets can introduce or amplify classification biases, especially for minority groups. In this work, we show that these biases can be exploited to increase the likelihood of privacy attacks against these groups. To do so, we propose a novel inference attack exploiting the knowledge of group fairness metrics during the training of the global model. Then to thwart this attack, we define a fairness-aware encrypted-domain aggregation algorithm that is differentially-private by design thanks to the approximate precision loss of the threshold multi-key CKKS homomorphic encryption scheme. Finally, we demonstrate the good performance of our proposal both in terms of fairness and privacy through experiments conducted over three real datasets.

Keywords

Federated Learning, Fairness, Fully Homomorphic Encryption, Privacy Attacks, Differential Privacy

1 Introduction

Machine Learning (ML) based systems are becoming pervasive in our connected society and have already led to countless practical applications, (*e.g.*, better health diagnosis or improved cyber-threat management). However, these systems are vulnerable to various privacy attacks [26]. Nonetheless, with growing public awareness and new legislations such as the GDPR[1], CCPA[2] or Privacy Act[3], significant efforts have been dedicated to address these privacy issues. In particular, collaborative approaches such as Federated Learning (FL) propose to tackle these privacy issues by making a set of clients, *a.k.a.* workers, train the same model without explicitly sharing their sensitive data. Each client locally trains the model on his private data. Then, all clients share their model updates with an aggregation server that creates the common model.

Even though clients' data are never shared directly with the server, they are still vulnerable to privacy attacks against the learned aggregated model [26, 45] as well as intermediary updates [69], while suffering from increased biases due to the imbalanced and non-diverse distribution of data.

Recent studies have also confirmed that privacy attacks are not uniform across groups, showing in particular around 20% more success for attacks against minority groups [14, 65].

The fairness-privacy duality has been explored in recent works in centralized learning [3, 14, 63]. However, reconciling these two ethical issues in decentralized training is even more challenging. In this work, we propose to study these two notions in FL settings, in which clients are often tasked with uploading extra information, along with the updated model, to enhance global group fairness.

Our work aims to demonstrate the privacy-sensitive nature of group-fairness local measures through an enhanced membership inference, emphasising also that existing works tend to enhance group fairness in FL at the expense of privacy. Furthermore, we show that approximate homomorphic encryption (*i.e.*, CKKS) can inherently provide Differential Privacy (DP) at no extra cost when evaluating a fairness-aware aggregation circuit over encrypted data.

To the best of our knowledge, this is the first work investigating the relationship between CKKS RLWE (Ring Learning With Errors) encryption noise and DP in the context of FL aggregation. The contributions of this paper can be summarized as follows:

- We propose a membership inference attack for FL exploiting the group fairness information[4]. Our proposed attack is based on synthetically generated data through Generative Adversarial

Proceedings on Privacy Enhancing Technologies 2025(1), 845–865

https://doi.org/10.56553/popets-2025-0044

[1] https://gdpr-info.eu/
[2] https://oag.ca.gov/privacy/ccpa
[3] https://www.legislation.gov.au/Series/C2004A03712
[4] Our attack code is available at https://github.com/Akram275/PETs_2024_99.git

Networks (GAN) for training shadow models [69, 84]. More precisely, the proposed GAN relies on two discriminators to impose dual constraints on the generated samples: closeness to real data distribution and fairness. The fairness constraint, weighted by a parameter λ, enables our shadow models to closely align with the fairness level of the target model. We show that considering unbalanced data improves the membership inference accuracy by around 20%, compared to the state-of-the-art.
- We introduce a privacy-preserving aggregation function achieving group fairness based on [30]. More precisely, to reconcile privacy and fairness considerations in the aggregation process within FL, we use CKKS, a Fully Homomorphic Encryption (FHE) scheme to hide the value of the fairness metric from honest-but-curious adversaries, while keeping the ability for the server to obliviously assign a weight to each client's update according to how fair his model is over his private dataset.
- We strengthen the privacy requirements through the use of a threshold multi-key version of CKKS algorithm to prevent single client decryption collusion, and prove the support of the differential privacy (DP) from approximate homomorphic aggregation precision loss by leveraging the key insights from the analysis presented in [18]. Specifically, we observe that ciphertexts' noise retains their Gaussian nature throughout homomorphic operations, establishing a link between homomorphic operations' precision loss and the DP via the Gaussian mechanism.
- We validate the proposed protocol through extensive experimental results, in which we quantify both the fairness level and utility of the final model.
  We collaboratively train models of various sizes, depending on the datasets, and demonstrate scaling up to two million parameters for the more challenging FairFace dataset. Our training approach implements an approximate FHE-based and fairness-aware aggregation algorithm, leading to a FHE computational overhead of only 5% as well as significant utility and fairness improvement of the final global model.

The paper is organized as follows. First, Section 2 provides an overview of algorithmic tools, and Section 3 reviews the prior research on privacy-preserving and fairness-aware FL. Afterwards, Section 4 describes the threat model and emphasizes the importance of hiding the fairness metric evaluation of a model through a novel membership inference attack. Then, Section 5 proposes an encrypted aggregation protocol to mitigate privacy issues through the assignment of a weight w.r.t. the quality of an update. Finally, Section 6 discusses the security of the proposed solution before describing and analyzing in Section 7 the conducted experiments.

2 Background

This section reviews the background notions concerning fairness, the CKKS FHE scheme as well as DP. Table 10 in Appendix A introduces different notations, and acronyms.

2.1 Fairness in Machine Learning

Overview. Algorithmic fairness investigates the behaviour of an algorithm with respect to inputs belonging to different groups defined by a sensitive attribute $\mathcal{S}$ that can lead to discrimination (*e.g.*, race or gender). In this context, group fairness refers to the statistical independence of the model's prediction $\hat{Y}$ from the sensitive attribute $\mathcal{S}$. Removing the sensitive attribute from the training data does not solve the problem since *indirect discrimination* can still occur due to other attributes that can act as *proxies* for the sensitive attribute, thus, reproducing the same discriminatory behaviour (*e.g.*, the zip code is highly correlated with ethnicity in USA, but remains relevant to predict the income).

In addition, artificially removing statistical correlations between sensitive and non-sensitive attributes is challenging, and often results in a fairness-accuracy trade-off [56].

Finally, the imbalance of the training as well as the presence of minority groups may lead to poor performance of the model over these groups.

Group fairness. Group fairness definitions seamlessly align with common societal discriminatory situations, and express the need for equality in the predictive performances of a model across different demographic groups. Furthermore, group fairness metrics, both on datasets and on classifiers trained on biased datasets, provide a comprehensive overview of the underlying data distribution.

Fairness-aware FL solutions require clients to periodically share these metric evaluations along with their updates, hence raising strong privacy concerns.

In addition, some works argue that individual fairness is a special case of group fairness, in which each group consists of a single record, and thus, group fairness solutions can be extended to individual fairness [93].

In the following, we discuss both data and classifier unfairness w.r.t. sensitive attribute $\mathcal{S}$. Without loss of generality, we consider that $\mathcal{S}$ is a binary attribute with values $\{s_0, s_1\}$.

Data unfairness. A labeled dataset may exhibit two distinct forms of discrimination expressed in the distribution of the joint features $(\mathcal{X}, \mathcal{S}, \mathcal{Y})$, in which $\mathcal{X}$ is the set of non-sensitive attributes and $\mathcal{Y}$ the class label.

- **Disparate Treatment** refers to the distribution $P(\mathcal{Y}|\mathcal{S})$, and therefore expresses the statistical correlations between the sensitive attribute $\mathcal{S}$ and the label $\mathcal{Y}$ within a dataset. It is often measured as a ratio between the proportion of positively labeled elements from group $\mathcal{S} = s_0$ and group $\mathcal{S} = s_1$:
$$DT(\mathcal{D}, \mathcal{S}) = \frac{P(\mathcal{Y} = 1|\mathcal{S} = s_0)}{P(\mathcal{Y} = 1|\mathcal{S} = s_1)}.$$
  Therefore, if $\mathcal{S} \perp \mathcal{Y}$ than $DT(\mathcal{D}, \mathcal{S}) = 1$.
- **Disparate Impact** extends the source of discrimination to non-sensitive attributes $\mathcal{X}$, considering them as statistical proxies to the sensitive ones. Hence, it models the distribution $P(\mathcal{S}|\mathcal{X})$. A common measure of disparate impact is quantified from the *Balanced Error Rate* (BER) [33] of the *best* performing adversarial classifier $\bar{f} : \mathcal{X} \to \mathcal{S}$ that infers the attribute $\mathcal{S}$ given $\mathcal{X}$. More precisely, BER satisfies:
$$BER(\bar{f}, \mathcal{S}) = \frac{P(\bar{f}(\mathcal{X}) = 1|\mathcal{S} = s_0) + P(\bar{f}(\mathcal{X}) = 0|\mathcal{S} = s_1)}{2}.$$
  A low $BER(\bar{f}, \mathcal{S})$ indicates a high correlation between $\mathcal{S}$ and $\mathcal{X}$, and therefore, high disparate impact in $\mathcal{D}$.

Classifier unfairness. Hereafter, we review the main group fairness metrics for evaluating the impact of data unfairness, with respect to a sensitive attribute, on a classifier's behaviour ($\hat{\mathcal{Y}}$).

- **Statistical Parity Difference (SPD)** [22] evaluates the proportion of positive outcomes across different groups defined by a sensitive attribute. For example, for profiles having the same qualification, the proportion of males and females hired should be roughly the same and satisfies: $P(\hat{\mathcal{Y}} = 1|\mathcal{S} = s_0) = P(\hat{\mathcal{Y}} = 1|\mathcal{S} = s_1)$. In this case, SPD will be equal to 0 as it satisfies:

$$SPD = |P(\hat{\mathcal{Y}} = 1|\mathcal{S} = s_0) - P(\hat{\mathcal{Y}} = 1|\mathcal{S} = s_1)|$$

  However, SPD does not take into account the model's accuracy. In particular, a dummy model that systematically outputs 1 ($h(x) = 1, \forall x$) would perfectly satisfy the statistical parity test.
- **Equal Opportunity Difference (EOD)** [37] ensures that the true positive rate is similar across different groups as defined by a sensitive attribute.

$$EOD = |P(\hat{\mathcal{Y}} = 1|\mathcal{S} = s_0, \mathcal{Y} = 1) - P(\hat{\mathcal{Y}} = 1|\mathcal{S} = s_1, \mathcal{Y} = 1)|$$

2.2 CKKS Cryptosystem

Overview. CKKS [18] is a homomorphic encryption scheme that allows computing additions and multiplications of ciphertexts, and relies on Ring-LWE [60] for encrypting messages encoded as polynomial ring elements. It encrypts vectors of complex numbers and supports homomorphic floating point arithmetic in the encrypted domain, with parametric precision, hence, perfectly suiting privacy-preserving machine learning tasks [53, 70, 77]. CKKS differs from other homomorphic encryption schemes (*e.g.* BFV [31], BGV [11] or TFHE [21]) regarding the interpretation of the native RLWE noise. Indeed, this scheme considers the encryption noise as a part of the message, as floating-point arithmetic for approximating real numbers. It considers the approximation to be sufficiently close to the exact value, such that $\mathsf{FHE.Dec}(f([m]_{\mathsf{FHE.pk}})) = f(m) + e_f$, in which e_f is the resulting approximation error. e_f depends on the encryption noise, and the multiplicative depth of the arithmetic circuit f. A detailed description of the CKKS building blocks, as initially introduced in [18], is provided in Appendix E.

CKKS noise-growth. The term "noise" refers to both the RLWE noise polynomial needed to ensure the security of the scheme, and to the precision loss during the approximate homomorphic computations. An informal definition of precision loss is the difference between the output of a plaintext evaluation of an arithmetic circuit and the decrypted output of the homomorphic evaluation of the same circuit over the same encrypted input(s).

The so-called "average noise" tracking approach provides a mechanism for tracking the stochastic properties of a ciphertext's noise (*e.g.*, its distribution, mean, and variance) as it progresses through each level of an arithmetic circuit evaluation. The Central Limit Theorem plays a crucial role in asserting that the noise maintains a Gaussian distribution throughout an entire arithmetic circuit in CKKS. In the following, we summarize the main results from [23], providing an average noise tracking of the CKKS scheme through homomorphic addition and multiplication, which is sufficient to characterize noise at the output of any homomorphic arithmetic circuit. The following theorem characterizes the nature of the noise component in a CKKS ciphertext after each transformation induced by homomorphic computations.

Theorem 2.1 (Noise distribution-preserving homomorphic operations). *Let $Z \sim \mathcal{N}(\mu, \rho^2 \mathcal{I}_N)$, and $Z' \sim \mathcal{N}(\mu', \rho'^2 \mathcal{I}_N)$ be two polynomials with coefficients sampled from the multivariate Gaussian distributions with respective covariance matrices $\rho^2 \mathcal{I}_N$ and $\rho'^2 \mathcal{I}_N$. The distribution of $Z + Z'$ ($modulo\, X^N + 1$), λZ ($modulo\, X^N + 1$) and the rounding coefficient-wise $\lfloor Z \rceil$ are given by:*

$$Z + Z' \sim \mathcal{N}(0, (\rho^2 + \rho'^2)\mathcal{I}_N), \quad \lambda Z \sim \mathcal{N}(\lambda\mu, ||\lambda||_2^2 \rho^2 \mathcal{I}_N)$$

$$and \quad \lfloor Z \rceil \sim \mathcal{N}(\mu, (\rho^2 + \frac{1}{12})\mathcal{I}_N).$$

Table 13, in Appendix E, summarizes the noise evolution through different homomorphic operations from [23].

2.3 DP and Gaussian Mechanism

Differential privacy, introduced by Dwork *et al.* [27], offers a mathematical framework for quantifying and ensuring the privacy of individual data samples in databases. It enables the extraction of global statistical information from a dataset while minimizing the risk of revealing sensitive information about any specific sample. More precisely, DP ensures that the output of the same randomized mechanism applied to two adjacent datasets $\mathcal{M}(\mathcal{D})$, and $\mathcal{M}(\mathcal{D}')$ (*i.e.*, identical except for a single entry) remains statistically indistinguishable while quantifying this indistinguishability.

Achieving DP usually involves employing perturbation techniques designed to *blur* the distinction between $\mathcal{M}(\mathcal{D})$, and $\mathcal{M}(\mathcal{D}')$, hence protecting individual private elements within a dataset $\mathcal{D}$ when observing $\mathcal{M}(\mathcal{D})$. This approach holds significant importance in the context of FL, especially when under a semi-honest server that may collude with clients and gain access to plaintext global model parameters θ_g^t for all $1 \leq t \leq T$. In supervised learning settings, perturbation is typically applied through several strategies:

- A first possibility is to add noise to the training data [27, 46, 47]. For instance, randomized response mechanisms can be interpreted as data perturbation and achieve differential privacy [89].
- Alternatively, noise can be introduced during the initialization of model parameters [46] or by incorporating additive noise into the objective function (loss function) that we aim to minimize [16].
- Lastly, noise can be added into the outputted model parameters, using Gaussian, Laplacian [27] or exponential mechanisms [28].

Definition 2.2 (l_p sensitivity). Let $f : \mathbb{R}^{k \cdot n} \rightarrow \mathbb{R}^d$ be a randomized mechanism operating on datasets represented as n real vectors of dimension k. The l_p sensitivity of f, denoted $\Delta_p f$ is:

$$\Delta_p f = max_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_p,$$

in which $\mathcal{D}, \mathcal{D}'$ are adjacent datasets.

Intuitively, the sensitivity quantifies how much the output of a mechanism can change when changing one row in the input dataset. In this work, and for convenience, we use the l_2 sensitivity, which will be referred to as "sensitivity", unless stated otherwise.

Theorem 2.3 (Gaussian mechanism & (ϵ, δ)-differential privacy [27].). *Let $f : \mathbb{N}^{k \cdot n} \rightarrow \mathbb{R}^d$ be a randomized mechanism with l_2 sensitivity Δf. The addition of zero-mean Gaussian noise to the output*

of f with variance $\sigma^2 = \frac{2ln(1.25/\delta)(\Delta f)^2}{\epsilon^2}$ provides (ϵ, δ)-differential-privacy to the ouput of f, which means:

$$\frac{Pr[f(\mathcal{D}) + \mathcal{N}(0, \sigma^2) = y]}{Pr[f(\mathcal{D}') + \mathcal{N}(0, \sigma^2) = y']} \leq exp(\epsilon) + \delta, \quad (1)$$

in which the probability is taken over all $(\mathcal{D}, \mathcal{D}')$ adjacent datasets, and the random behaviour of f.

When f is a supervised learning process, the randomness is mainly due to the sampling of the data batches, as well as the parameter initialization.

3 Related Work

3.1 Privacy-preserving and Fairness-aware FL

FL offers a privacy-preserving alternative to traditional training methods by keeping data locally on clients' devices. However, the leakage of sensitive information can still occur through intermediate clients' updates and the aggregated model [29, 55, 65, 78]. Thus, FL still requires the use of privacy-enhancing technologies for securing the aggregation process. Existing privacy-preserving solutions for FL target mainly two adversary models: honest-but-curious (or semi-honest) adversaries and malicious adversaries [25]. To address the honest-but-curious adversaries threat, several solutions implement techniques for secure aggregation [58, 66], including FHE [8, 74, 86], functional encryption [82], MPC [12], pairwise masking [61, 85], along with DP [58, 86, 92].

FL can improve fairness by leveraging the statistical diversity offered by the distributed training datasets but also raises several challenges. For instance, Chang and Shokri [15] have observed that in FL, unfairness propagates from clients with highly biased data to clients with fairer data. Zeng *et al.* [94] have shown that fair classifiers in the FL setting usually display a lower performance than the ones in the centralized setting. They proposed FedFB (Federated FairBatch) as an alternative to FedAvg [64]. FedFB dynamically adjusts the batch sizes for different groups to improve the fairness of the global model. Rodriguez *et al.* [79] have tackled the issue of DP's unfair impact on a classifier's performance over under-represented groups with differential multipliers [75]. Ezzeldin *et al.* [30] have designed `FairFed` a fairness-aware aggregation, which re-weights clients' updates with respect to their local fairness metrics. It improves the global model's fairness compared to `FedAvg`.

3.2 Tensions between Privacy and Fairness

In centralized learning. Recent works show the existence of significant tensions between fairness and privacy in ML [3, 14, 17, 65, 72]. The impact of implementing DP on fairness has also been studied in [3, 14, 24, 35]. Previous works have demonstrated an accuracy degradation due to the implementation of DP being more significant for minority groups [6]. Pujol *et al.* have investigated why these groups are disproportionately impacted while also proposing solutions to improve the fairness of the resulting model [76]. Jagielski and collaborators [42] have proposed two learning algorithms that aim to fulfill fairness, DP, and accuracy requirements by exploring their trade-offs.

Privacy attacks and fairness. We distinguish two categories of works investigating the relationship between the group fairness of a model and its vulnerability against privacy attacks.

- **Impact of fairness interventions on MIA vulnerability:** Kulynych *et al.* [49] have studied the disparate vulnerability of MIA attacks, showing that fairness interventions could reduce disparate vulnerability and DP training limits the extent of this vulnerability. Tian *et al.* [87] have evaluated three standard MIAs [57] against binary classifiers trained with and without fairness interventions, demonstrating a significant accuracy degradation of most attacks against the fair classifiers. While the scores tend to increase for member data, they tend to align with a normal distribution for non-member data after fairness interventions.
  However, these works are limited to empirically measuring the impact of fairness interventions on a model's membership vulnerability.
- **Fairness information as an auxiliary knowledge in privacy attacks:** Ferry *et al.* [34] have proposed a generic approach by which the knowledge of a fairness metric can enhance the accuracy of an attribute inference attack by setting the fairness metric value as an additional constraint to be satisfied by the adversary's guesses.

In federated learning. Fairness-aware FL methods [30, 32, 39] can be subject to several privacy issues. For instance, Chen *et al.* [17] have discussed the main FL privacy concerns and existing approaches to address the fairness-privacy trade-off. Padala *et al.* [71] have studied the combination of local DP and group fairness.

More precisely, in their framework each client trains a fair and accurate model using its local dataset before later learning a surrogate model to align the fair predictions from the initial model with a guaranteed DP at the cost of an increased computation overhead. However, building the differentially private surrogate introduces a significant local computational overhead.

Uniyal *et al.* [88] have compared two algorithms for training deep neural networks, namely DP-SGD and PATE, to analyze their fairness-privacy trade-off, concluding that PATE [73] produces student models with improved group fairness compared to DP-SGD [1]. Zhang *et al.* [95] have designed a secure aggregation that aims to reduce bias in FL applications that are privacy sensitive. Whether a client participates in a given round of training is based on the current unfairness level of the global model and whether his local data improves the global model's fairness. Hence, the fact that clients either participate or not in the training leaks information on their local data distribution. Ruckel *et al.* [81] have combined blockchain, local DP, and zero-knowledge proofs (ZKP) to achieve privacy, clients' fairness, and integrity. Nevertheless, the induced computational overhead from integrity verification is significant. Rodríguez-Gálvez *et al.* [79] extended the modified method of differential multipliers [75] to empirical risk minimization with fairness constraints, thus providing a centralized fairness-aware training algorithm, for which they provide a decentralized version, Hence improving group fairness under differential-privacy constraints.

Tables 1 and 2 provide a comprehensive comparison of our proposed solution against closely related work. More precisely, Table 1 compares each approach's security and privacy properties under

different threat models while Table 2 displays the theoretical performances in terms of computation and communication overheads. Our proposed solution consistently outperforms closely related work, particularly in maintaining robust privacy protections while considering group fairness, under the honest-but-curious threat model with colluding entities. In summary, our work either outperforms other approaches regarding computational and communication complexities or jointly integrates privacy and fairness considerations not considered by these previous works.

4 Fairness-enhanced inference attack

4.1 Adversary model

We consider a FL setting in which n entities collaborate to obtain a global shared model with the support of a central server. To evaluate the security of the proposed framework, we consider honest-but-curious (or semi-honest) adversaries for both the aggregating server and the participating clients. Some clients and even the server may collude. They will not deviate from the protocol as they are semi-honest, but they will attempt to use and merge their respective knowledge to infer extra information about other clients' updates or datasets. We will refer to the colluding entities as passive attackers. Our framework only requires the knowledge of a bound B_{passive} on the number of potentially passive attackers ($0 \leq B_{\text{passive}} < n$). Additionally, the successive approximate homomorphic aggregations with CKKS introduce additive noise to the global model, which we show to exhibit characteristics of a Gaussian mechanism. Hence, this provides a level of DP guarantees.

4.2 Membership Inference Attack from Fairness Information

A standard approach for conducting a membership inference attack is to train an attack model that predicts x as member or non-member according to the behaviour of the classifier $\mathcal{M}$ when presented with x (*e.g.*, prediction confidence, logits or loss).

We follow the membership attack approach by Shokri *et al.* [84], and include the knowledge of a fairness metric evaluation of the target model. Hereafter, we extend this attack framework by including additional key information regarding the model's behavior to imitate. Specifically, we emphasize that imitating the unfairness level of the target model by the set of shadow models in [84] enhances the utility of the attack dataset (closer predictions to the ones of the target model), and thus, improves attack classifier's performance. This highlights the privacy sensitivity of these measures.

Attack overview. The key steps of the attack are the following:

(1) When provided as input a target model's $\mathcal{M}$ fairness metric evaluation f_{target}, a synthetic data generation monitored by f_{target} is conducted, which produces a dataset containing a level of bias proportional to f_{target}.
(2) k Shadow models $(S_i)_{i\in[k]}$ are trained using non-overlapping portions of the previously generated data to exhibit similar unfair behaviour w.r.t. the fairness metric: $F_{S_i} \approx f_{\text{target}}$.
(3) The attack dataset consisting of inference of the previously trained shadow models on member and non-member data points and their respective membership status is formed.
(4) Finally, the attack model is trained using the attack dataset.

The first step of the attack is particularly crucial as generating biased data in proportion to a specified parameter is a challenging task. Indeed, the behavioural closeness of the shadow models to the target model has a major impact on the attack model accuracy. For instance in [84], the authors assume that the adversary has a collection of data following the same distribution as the target model's training data. Another possibility to train the shadow models is the use of synthetic data. However, most synthetic data generation methods are primarily designed to closely match the training data distribution rather than intentionally introducing tailored bias. Figure 1 summarizes the different steps of the attack.

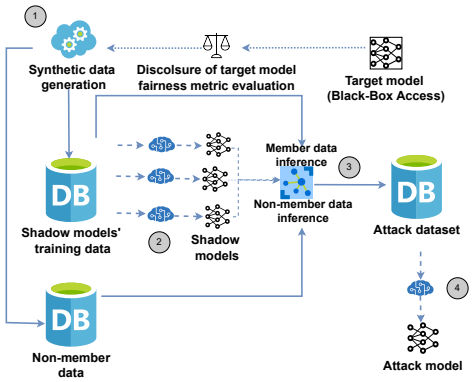


Figure 1: Enhanced membership inference attack by exploiting fairness metric disclosure.

Unfairness in synthetic data. To improve the membership inference attack in [84], it is important to understand the process of unfairness monitoring. The shadow classifiers' fairness is influenced by how discrimination is parameterized within synthetic data. There are two main types of discrimination: disparate treatment and disparate impact. To monitor unfairness in a dataset $\mathcal{D} = (\mathcal{X}, \mathcal{S}, \mathcal{Y})$, control over both $P(\mathcal{Y}|\mathcal{S})$ and $P(\mathcal{S}|\mathcal{X})$ is necessary.

Xu *et al.* [90] have designed FairGAN [5]; a GAN with a generator network G_{Dec} and two discriminator networks D_1 and D_2. D_1 ensuring that the distribution P_G of data generated by G_{Dec} is as close as possible to the real data distribution P_{data} as done in classical GAN architectures. Meanwhile, D_2 distinguishes between the two conditional distributions $P_G(\mathcal{X}, \mathcal{Y}|\mathcal{S} = 1)$ and $P_G(\mathcal{X}, \mathcal{Y}|\mathcal{S} = 0)$ within the generated samples. Hence, D_2 acts as an adversarial model predicting $\mathcal{S}$ from $\mathcal{X}$ and $\mathcal{Y}$. Consequently, the objective of G_{Dec} is twofold: (1) fooling D_1 by making P_G indistinguishable from P_{data} and (2) cheating D_2, by making $P_G(\mathcal{X}, \mathcal{Y}|\mathcal{S} = 1)$ as close as possible to $P_G(\mathcal{X}, \mathcal{Y}|\mathcal{S} = 0)$, thus reducing both disparate impact and disparate treatment. More precisely, the loss $L_{(G_{Dec}, D_1)}$ of the generator w.r.t. D_1 expresses the capability of G_{Dec} to produce seemingly genuine samples. Meanwhile, $L_{(G_{Dec}, D_2)}$ is the loss of G w.r.t. D_2. It reflects the ability of G to produce samples with indistinguishable sensitive attribute given the rest of the attributes and the label. Finally G_{Dec} is trained to minimize the expression $L_{(G_{Dec}, D_1)} + \lambda L_{(G_{Dec}, D_2)}$.

[5] Further details on the FairGAN architecture are provided in Appendix B.

Table 1: Comparison between our solution and other state-of-the-art approaches with respect to fairness and privacy requirements.

	Threat Model		Privacy Requirements		Fairness Metrics	
	Honest but Curious	Malicious	Conf. of the updates	Conf. of the agg. model	Group Fairness	Individual Fairness
[91]	✓	✗	✓(Functional Encryption)	✗	✗	✗
[71]	✓(no collusions)	✗	✓(DP)	✓(DP)	✓(EOD)	✗
[94]	✓	✗	✓(DP)	✗	✓(EQD/SPD)	✗
[86]	✓	✗	✓(Additditive HE)	✓(DP)	✗	✗
[95]	✓	✗	✓	✗	✓(TPR TNR difference)	✗
[62]	✓	✓(Verifiable Computing)	✓(Additditive HE)	✗	✗	✗
[79]	✓	✗	✓(DP)	✓(DP)	✓(FNR/ACCURACY) parities	✗
Proposed	✓(with collusions)	✗	✓(FHE)	✓(FHE + DP)	✓(EOD/SPD)	✗

EQO denotes the Equalized Odds fairness metric, TNR refers to True Negative Rate and TPR refers to True Positive Rate (also known as Recall).

Table 2: Theoretical analysis of the computational and communicational complexities.

	Proposed	[91]	[71]	[94]	[86]	[95]	[62]	[79]
Datasets	*Adult-Census-Income* *Compas-Recidivism* *FairFace*	MNIST *OptDigit* *FairFace*	*Adult-Census-Income* *Bank*	*Adult-Census-Income* *Compas-Recidivism* *Dutch*	MNIST CIFAR-10	Adult-Census-Income *Compas-Recidivism*	FEMNIST	*Adult-Census-Income* FEMNIST
Comp. Cost	*Client :* $O(n)+$ TLT *Server :* $O(T(nd+n))$	*Client :* TLT *Server :* $O(T(nd+n))$	*Client :* 2TLT *Server :* $O(T(nd+n))$	*Client :* TLT *Server :* $O(T(nd+n))$	*Client :* $O(T(dN+nlogn))+$ TLT *Server :* $O((dn+dN+nlogn)T)$	*Client :* $O(T)+p$TLT *Server :* $O(T(nd+n))+O(T)$	*Client :* LT + TagGen *Server :* $O(T(md+m))$ + Sign	*Client :* TLT + $O(1)$ *Server :* $O(T(md+m))$
Com. Cost	*Client :* $O(n^2+dT)$ *Server :* $O(dnT)$	*Client :* $O(dT)$ *Server :* $O(dnT)$	*Client :* $O(dT)$ *Server :* $O(dnT)$	*Client :* $O(dT)$ *Server :* $O(dnT)$	*Client :* $O((d+n+N)T)$ *Server :* $O((dn+N)T)$	*Client :* $O(pdT))$ *Server :* $O(dnT)$	*Client :* $O(dT)$ *Server :* $O(dmT)$ N.A	*Client :* $O(dT+T)$ *Server :* $O(dnT)$

n, d and T are respectively the number of clients in the FL setting, the update dimension (*i.e.*, model or gradients size) and the number of iterations. When m is used instead of n, this indicates client subset sampling $m \leq n$. LT denotes local training cost, therefore TLT indicates total local learning cost. Sign and TagGen indicate the overhead of the verifiable mechanism operations in [62]. N denotes the dimension of the LWE masking used in [86] secure aggregation's approach. For [61], L denotes the bound on tolerated client dropouts and A is an upper bound on the number of neighbors of a client (for a PRG pairwise masking mechanism where pairs of clients secretly share their seeds). N.A denotes complexities that are expressed in a more complex case-by-case analysis. Regarding [95] p indicates the proportion of rounds at which each client participates.

The process of unfairness monitoring is ensured by the scaling factor λ. Setting $\lambda = 0$ removes the fairness constraint and transforms FairGAN into a regular GAN achieving only $P_G \approx P_{\text{data}}$. It outputs synthetically generated data with a baseline unfairness level f_0 approximately equal to the training data's unfairness. Larger values of λ linearly improve fairness as shown in Figure 2. From our experiments (up to $\lambda = 1.7$), the following relationship between the achieved fairness f, and λ is satisfied, such that $f \approx \frac{-2}{5}\lambda + f_0$ with $\lambda \in [0, 1.7]$.

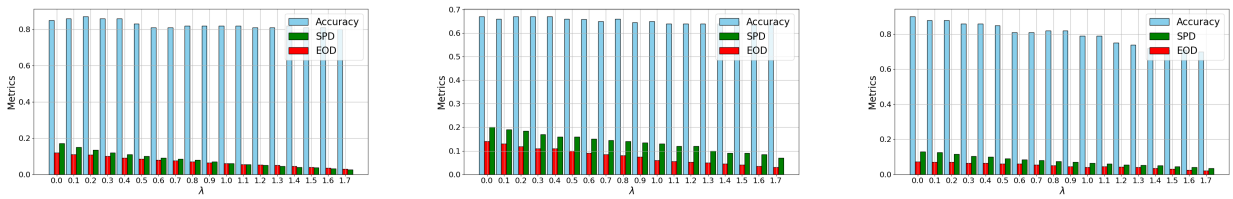
Thus, to achieve a target fairness measure f_t, the attacker evaluates its shadow training data's fairness f_0 while parameterizing the dual GAN with $\lambda = \frac{-5}{2}(f_t - f_0)$. One limitation of our approach is that an attacker cannot achieve a fairness level that is worse than f_0 (*i.e.*, $f_t > f_0$). Indeed, negative values of λ yield unstable GAN training and unpredictable fairness levels. Consequently, highly unfair shadow training data are preferable, since they allow adversaries to achieve unfairness levels within larger intervals.

Experiments and analysis. We have implemented the FairGAN architecture, adopting a fully connected neural network generator featuring two hidden layers of 128 and 125 units, with a BatchNormalization layer in between. The generator architecture after iterative adjustments of hyper-parameters and dimensions, operates in a latent (*i.e.*, noise) space of dimension 100, producing samples in an intermediate space of dimensions 75 for the *Adult* samples and 50 for the Compas samples. These lower-dimensional representations are transformed into synthetic data samples using the decoder component of a pre-trained autoencoder, which is composed of a single hidden layer in each of the encoder and decoder halves. Regarding the two discriminators, we select classical architectures that can provide an optimally performing classifier on these datasets (to predict income and recidivism). This corresponds to fully connected neural networks with two hidden layers of 128 and 64 units. The autoencoder is trained on the original datasets for 500 epochs, before incorporating its decoder half to the generator to build the G_{Dec} component. In contrast, the GAN architecture is first trained without D_2 for 1000 epochs, because the first samples produced by the G_{Dec} component are meaningless, and therefore, searching for unfairness patterns within these records might perturb the training process. Afterward, D_2 is integrated into the architecture for extra training epochs until the network reaches a stable state (around 1500 additional training epochs). Overall, the training process of the synthesizer for *Adult*, and Compas aligns with the methodology outlined in [90], proving to be effective in achieving notable results.

The membership inference attack is conducted through the implementation of the shadow training available in *Adversarial Robustness Toolbox* [6]. The attack is tested with multiple datasets for training the shadow models, which are generated using multiple trained FairGAN networks with different values of λ ranging from 0 to 1.7. The attack performances are reported over 50 attack classifiers from shadow models trained on various synthetic data from FairGAN architecture with different values of λ. Figure 3 depicts the performance measures of all attack classifiers given the proximity of their respective shadow models average fairness metric to the target model fairness (EOD and SPD): $\frac{\sum_{i=1}^{n_{sh}} F(S_i)}{n_{sh}} - F(\mathcal{M})$, with $F \in \{\text{EOD}, \text{SPD}\}$.

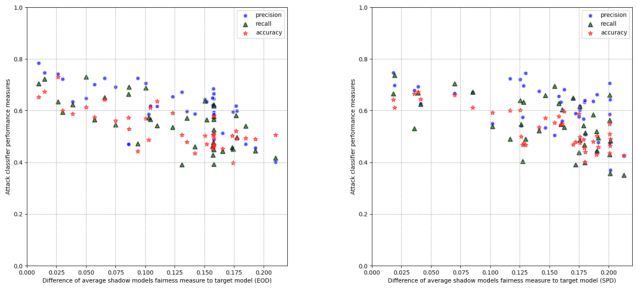
Controlling the fairness level enhances the membership attack accuracy by around 19% (from 0.5 to 0.63) when the considered metric is EOD and 15% when the metric is SPD. Overall, the EOD metric is more dependent on the data distribution than SPD as it is more sensitive to disparate impact within training data.

[6] https://github.com/Trusted-AI/adversarial-robustness-toolbox



Adult **Compas** **FairFace**

Figure 2: Classifiers accuracy and fairness (SPD and EOD) for the groups "Male/Female" trained on synthetic datasets (20k samples) generated using FairGAN with different λ (x-axis) and measured on the original *Adult* Dataset (32k samples).



EOD proximity **SPD proximity**

Figure 3: Membership attack accuracy on a classifier trained on *Adult* dataset, for various fairness measures differences between the average of shadow models' fairness measures and the target model's fairness measure.

4.3 Weight-based Privacy Leakage Evaluation

In the following, we evaluate the relation between the weight ω_i assigned to a local update θ_i^t from a client i (whose value belongs to $[0, 1]$) and the privacy leakage associated with his dataset. We consider the scenario in which two clients with their datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ upload their model updates θ_1^t and θ_2^t to the server.

Predictions' distances. We further observe the distances between a prediction vector $\hat{y} = M_{\theta_g^{t+1}}(\mathcal{D})$ made by the aggregated model on a set of data samples $\mathcal{D}$ at iteration t, and the two prediction vectors from the pair of models. We have $\hat{y}_1 = M_{\theta_1^{t-1}}(\mathcal{D})$ and $\hat{y}_2 = M_{\theta_2^{t-1}}(\mathcal{D})$. Figures 4 show the Euclidean distance between $\hat{y}$ and $\hat{y}_1$ according to its aggregation weight.

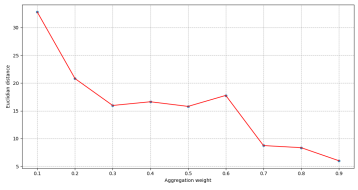
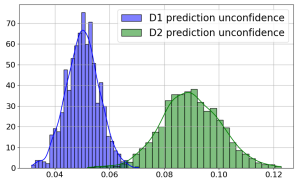


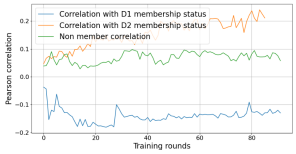
Figure 4: Euclidean distance $\|\hat{y} - \hat{y}_1\|_2$ as a function of ω_1

Experiments' analysis. We analyze the vulnerability of each client's dataset to membership attacks against a global model trained with FL (FedAvg). We refer to *prediction unconfidence* as the difference between the raw predictions (*i.e.*, the sigmoid output), and the predicted class. In the case of a binary classifier, the prediction unconfidence is $\lceil M_{\theta_g}(x) \rfloor - M_{\theta_g}(x)$. The prediction unconfidence is represented as an array PC computed over the predictions on the union of the two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$. Three binary arrays represent the membership status of the two datasets: $A_1[i] = \{1 \text{ if } x_i \in \mathcal{D}_1, 0 \text{ } Otherwise\}$, $A_2[i] = \{1 \text{ if } x_i \in \mathcal{D}_2, 0 \text{ } Otherwise\}$ and $A_3[i] = \{1 \text{ if } x_i \notin \mathcal{D}_1 \cup \mathcal{D}_2, 0, \text{Otherwise}\}$.

For our analysis, we first observe the prediction unconfidence evaluation of the global model over data samples from different distributions of $\mathcal{D}_i$ with distant values of ω_i and ω_2. Then, we evaluate the Pearson correlations between the global model's prediction unconfidence on data samples from different sources, and the binary vectors A_1, A_2, and A_3, representing respectively the membership status of $\mathcal{D}_1$, $\mathcal{D}_2$, and non-member data.



Prediction unconfidence distributions of global model on data from $\mathcal{D}_1$ and on data from $\mathcal{D}_2$ after 200 learning iterations with constant weights $\omega_1 = 0.9$ and $\omega_2 = 0.1$



Evolution of Pearson's correlation between predictions unconfidence and membership status of $\mathcal{D}_1$, $\mathcal{D}_2$, and non-member status, with constant weights $\omega_1 = 0.9$ and $\omega_2 = 0.1$.

Figure 5: FL trained classifier's behavior with respect to data samples' membership status and aggregation weights.

The main observation from Figure 5 is the existence of a per-dataset overfitting phenomenon. Indeed, Pearson correlations show a strong relationship between the weight assigned to an update and the prediction (un)confidence for data samples in the dataset used to compute those updates. The final global model behaves almost similarly when given data samples from small-weighted datasets ($\omega_2 = 0.1$) or non-member data samples (neither in $\mathcal{D}_1$ nor $\mathcal{D}_2$).

Therefore, instead of a single-bit information attack (member/non-member), this observation opens the perspective for membership attacks on a final aggregated model with the location of the membership of a data sample (member at $\mathcal{D}_i$). This analysis reveals that an aggregation server that is granted access to the aggregation weights or a metric value proportional to them can establish a ranking of vulnerable clients' datasets. This raises many issues: the server would focus the privacy attacks on the highest-weighted clients and clients would have minimal incentive to improve the quality of their updates, as the improvement is linked to the extent of privacy exposure they experience. Typically, in a fairness-driven aggregation strategy, clients are encouraged to improve their updates by local data pre-processing. Hence, additional privacy guarantees regarding the assigned weights should be considered.

5 Privacy-preserving and Fairness-aware Federated Learning

We want to protect the privacy of individual updates, encompassing trained models on local data and their respective fairness metrics. To do so, we describe a privacy-preserving aggregation algorithm for the `FairFed` framework [30]. We then show how to enhance the security of `FairFed` by homomorphically aggregating the different updates. For this purpose, we rely on CKKS and provide the support of DP guarantees after decryption.

5.1 FairFed Description

The `FairFed` framework was proposed by Ezzeldin *et al.* in 2021 [30]. It enables group fairness in a federated learning setting, by performing the following steps at every iteration of the training:

(1) Each client i computes a model update θ_i^t and evaluates its fairness metric F_i^t (*i.e.*, EOD or SPD) on his test data (local train/test partitioning). Then, he shares θ_i^t and F_i^t with the aggregation server.
(2) Upon receiving the set of model updates $\{\theta_1^t, \ldots, \theta_n^t\}$ (n is the number of clients), and their respective fairness metrics $\{F_1^t, \ldots, F_n^t\}$, the server computes for every update θ_i^t the associated weight ω_i^t from F_i^t as:

$$\omega_i^t = \frac{\hat{\omega}_i^t}{\sum_{i=1}^n \hat{\omega}_i^t}, s.t: \ \hat{\omega}_i^t = \frac{n_i}{\sum_{k=1}^n n_k} \cdot exp(-\beta \cdot |F_g - F_i^t|)$$

where F_g is the global fairness evaluation (over the union of clients datasets) computed by aggregating locally computed and transmitted statistical measures (denoted $m_{global,k}$ in [30]). We note that $\frac{n_i}{\sum_{k=1}^n n_k}$ is scaled by a value in $[0, 1]$ proportional to the closeness of F_i^t to F_g. Indeed, exp will tend to 1 if F_i^t is close to F_g, else it will tend to 0. That is, the final ω_i^t will tend to $\frac{n_i}{\sum_{k=1}^n n_k}$ if F_i^t is close to F_g, else it will tend to 0.
(3) Finally, the server computes the aggregated model at iteration $t+1$ as $\theta_g^{t+1} = \sum_{i=1}^n \omega_i^t \cdot \theta_i^t$. As for any client i, ω_i^t will tend to 0 if F_i^t is not close to F_g, only clients with a fair model will participate to θ_g^{t+1} with their θ_i^t.

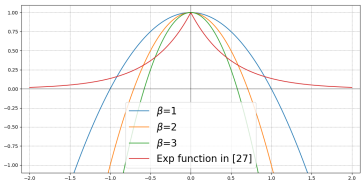


Figure 6: Polynomial alternatives for $exp(-\beta|F_i^t - F_g|)$ with $\beta \in \{1, 2, 3\}$

5.2 Enabling Private and Fair Model Training

In this section, we provide a privacy-preserving alternative to `FairFed` aggregation [30]. We propose to make each client i encrypt his model update θ_i^t and fairness metric F_i^t with CKKS before their transmission to the aggregation server. As such, the server will compute the aggregation on encrypted data. During the `FairFed` aggregation, as F_i^t approaches F_g, $exp(-\beta|F_g - F_i^t|)$ tends to 1, and the weight assigned to θ_i^t gets closer to $\frac{n_i}{\sum_{k=1}^n n_k}$. In contrast, the further F_i^t is from F_g, the closer ω_i^t becomes to 0. The homomorphic unfriendliness of this function is due to:

(1) Computing exp on encrypted data requires polynomial approximations. A good polynomial approximation of exp on the interval $[-1, 1]$ involves high degree polynomials (*e.g.*, $1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}$), implying larger multiplicative-depth [7], and bigger parameters for CKKS, increasing the FHE overhead.
(2) The scaling approach introduced in [30] would require a homomorphic computation of the absolute value function, which is challenging in FHE. Indeed, it requires a homomorphic sign function, often approximated by high-degree polynomials [51].

They ensure similar logic to the use of exp on the difference $|F_i^t - F_g^t|$. However, they provide a marginally different weight decrease. As the discrepancy between F_i^t and F_g^t widens, the exp function imposes a steep penalty on small values of $|F_i^t - F_g^t|$. This penalty gradually diminishes as the two measures diverge, preventing an excessive decline in the corresponding update utility. In contrast, the considered polynomial functions slightly penalize small discrepancies between F_i^t and F_g^t but abruptly decrease in value when the two measures exhibit significant divergence. Fortunately, a careful choice of β allows to linearly mimic the effect of the weight assignation mechanism of `FairFed`. Furthermore, degree 2 polynomials can be used to avoid the homomorphic evaluation of the absolute value, which would have involved the homomorphic evaluation of the *sign* function through polynomials approximations [51], thereby returning to the original non-linearity challenge. Appendix G provides an analysis of polynomial alternatives for the weight assignment mechanism of [30].

After selecting a linear alternative for exp, the issue of divisions remains. While normalization is employed in `FairFed` involving a division by the sum of pre-normalized weights, FHE renders it intractable, as dividing ciphertexts is infeasible. We propose to delegate the normalization to the clients, who perform it in cleartext after receiving the encrypted and aggregated model from the server. With division being distributive over the sum, the server broadcasts

[7] The total number of sequential homomorphic multiplications that can be performed on a fresh ciphertext with respect to the FHE scheme's parameters [11].

encryptions of $\theta_g^{t+1} \cdot \sum_{k=1}^{n} n_k$ and $\sum_{k=1}^{n} n_k$ to the n clients who perform collaborative decryption of the latter elements, followed by plaintext division to derive θ_g^{t+1}.

Table 3: Numerical examples of the polynomial alternative to exp with $\beta = 1$

$F_i^t - F_g^t$	$exp(-\beta\lvert F_i^t - F_g^t\rvert)$	$-\beta(F_i^t - F_g^t)^2 + 1$
±0.10	0.90	0.99
±0.25	0.78	0.94
±0.5	0.60	0.75
±0.75	0.47	0.43
±0.90	0.40	0.19

5.3 Framework Workflow

Having overcome the challenges associated with adapting `FairFed` aggregation for the encrypted domain, we now describe the steps for enabling private and fair federated learning model training.

(1) Clients collaboratively derive homomorphic global and partial secret CKKS keys, following the generic threshold key setup protocol of Mouchet *et al.* for RLWE-based schemes [67] with $t = B_{\text{passive}} + 1$ (Details in Appendix E.2). Each client i holds the public key pk and a share of the secret key sk_i, collaboratively derived with the participating entities during the setup phase. The evaluation key evk is transmitted to the server.
(2) Each client i encrypts θ_i^t and the associated F_i^t using his sk_i and sends them to the aggregating server.
(3) Upon receiving the encrypted updates and the associated fairness metrics evaluations of the participating entities, the server homomorphically computes encryptions of the pre-normalized weights $\{\hat{\omega}_1^t, \ldots, \hat{\omega}_n^t\}$, as detailed in section 5.2.
(4) The server shares the resulting encrypted aggregated model with the clients, along with the sum of the pre-normalized weights $\sum_{i=1}^{n} \hat{\omega}_i^t$ for a collaborative t-out-of-n decryption phase.
(5) Once the plaintext scaled global model parameters $\sum_{i=1}^{n} \hat{\omega}_i^t \cdot \theta_g^{t+1}$, and the sum $\sum_{i=1}^{n} \hat{\omega}_i^t$ are recovered from the collaborative decryption, global model parameters for iteration $t+1$ are obtained by plaintext division, and clients proceed to the next iteration of the learning procedure.

5.4 FHE vs Pairwise-masking for Privacy and Fairness in FL

Pairwise-masking allows the computation of the sum of confidential values held by clients without revealing them. The core idea involves using additive masks on the messages, in which the masks' modular sum cancels out to zero. Hence, it has been widely adopted as a secure aggregation mechanism [7, 43, 66, 83]. However, when a more elaborated aggregation strategy is involved, threshold FHE is preferred for the following reasons:

(1) The masking mechanism does not inherently support scaling by server-computed values. Therefore, implementing a fairness-aware aggregation strategy relying on privacy-sensitive fairness scores to determine aggregation weights would not be possible using the naive pairwise-masking approach [8]
(2) Using one-time pads requires a setup at each FL training round, adding significant computational and communication overhead in pairwise masking. While the use of Diffie-Hellman agreed-on seeds with a Pseudo-Random Number Generator mitigates this limitation [9], the setup must be repeated whenever the client set changes [66].
(3) From a privacy viewpoint, in contrast to FHE-based aggregation, the server obtains the plaintext aggregated model at every iteration of pairwise-masking-based FL. Without DP guarantees, the aggregated model can be subject to substantial privacy-leakage [29]. Thus, to protect the privacy of the aggregated model, a Distributed Differential Privacy (DDP) protocol would need to be implemented [40, 44]. This in turn raises other challenges, in particular with respect to honest-but-curious clients colluding to eliminate a large portion of the DP noise.

6 Security and Privacy Analysis

This section discusses the DP brought by CKKS approximations and analyses the resistance of our framework against the different adversaries, along with the main challenges induced by CKKS.

6.1 DP from CKKS Aggregation

The homomorphic evaluation of the aggregation function in Section 5.3 using the CKKS scheme from homomorphically computed weights over fairness measures, and encrypted updates yields :

$$\mathsf{FHE.Dec}_{\mathsf{FHE.sk}}\Big(\sum_{i=0}^{n} [\omega_i^t]_{\mathsf{FHE.pk}} \cdot [\theta_i^t]_{\mathsf{FHE.pk}}\Big) = \theta_g^{t+1} + e_{\text{agg}}$$

We follow the methodology introduced in [70] in which the authors estimate DP guarantees for a ridge regression training on CKKS encrypted data. Indeed, in this work, we establish that the approximation noise e_{agg} is Gaussian and centered and estimate its variance, before finally bounding the sensitivity of the mechanism subject to the Gaussian noise. These elements enable us to derive bounds for the associated (ϵ, δ)-differential privacy obtained from the approximate homomorphic fairness-aware aggregation.

Lemma 6.1. *A homomorphic aggregation with encrypted updates, and weights computed from encrypted fairness measures using the CKKS in threshold mode (c.f., appendix E.2), provides a Gaussian mechanism (from the approximate homomorphic computations) with variance:* $\sigma_{agg}^2 = n[\frac{1}{\Delta^2}(\sigma_{model\_scale}^2 + \sigma_{relin}^2) + \sigma_{round}^2]$ *, such that:*

- Δ *is the scaling factor (i.e., a CKKS parameter described in App. E)*
- n *is the number of clients updates*
- $\sigma_{model\_scale}^2$ *denotes the variance induced by the scaling of an encrypted update by the computed weight from the transmitted fairness metric* F_i^t *and satisfies:*

$$\sigma_{model\_scale}^2 = N\sigma_{fresh}^2\sigma_{fairness}^2 + \sigma_{fresh}^2\|\theta_i^t\|_2^2 + \sigma_{fairness}^2\|\omega_i^t\|_2^2$$

where :
– ω_i^t *is client i's assigned update weight*

[8] Scaling the masked updates results in scaling the masks (M_i^t). The scaled masks' sum does not cancel out: $\sum_{i=1}^{n} \omega_i^t(\theta_i^t + M_i^t) = \theta_g^{t+1} + \sum_{i=1}^{n} \omega_i^t M_i^t$ and $\sum_{i=1}^{n} \omega_i^t M_i^t \neq 0$.

– $\sigma^2_{fresh} = (\frac{4}{3}N + 1)\sigma^2_{init}$ *is the noise variance of a fresh ciphertext encrypting of the model's parameters using the standard* $\|v\|_2^2 = \frac{2N}{3}$, *and* $\|s\|_2^2 = \frac{2N}{3}$ *[18].*
– $\sigma^2_{fairness}$ *is the noise variance of the ciphertext encrypting the weight obtained from the encrypted fairness metric* $[F_i^t]$ *when the degree 2 linear alternative to exp is used with parameter* β*:*

$$\sigma^2_{fairness} = \beta^2 2N\sigma^2_{fresh} + 4\sigma^4_{fresh}\|F_i^t - F_g^t\|_2^2$$

Proof. The ciphertext encoding, addition, and multiplication preserve the Gaussian nature of the additive RLWE noise [23] ($e_{\text{agg}} \sim \mathcal{N}(\mu_{\text{agg}}, \sigma^2_{\text{agg}})$) . Hereafter, we characterize μ_{agg}, and σ^2_{agg}

- Mean: our computation involves (1) a plaintext/ciphertext addition ($[F_i^t]_{\mathsf{FHE.pk}} - F_g^t$) with no impact on the $[F_i^t]_{\mathsf{FHE.pk}}$ noise mean, so remaining zero-centered as sampled at the encryption phase. (2) The degree 2 polynomial evaluation on the ciphertext ($[F_i^t - F_g^t]_{\mathsf{FHE.pk}}$) requires ciphertext multiplication, scalar multiplication, and ciphertext/ciphertext additions. These operations preserve the zero-mean noise distribution. Hence the noise of the ciphertext encrypting the non-normalized weights follows a zero-mean Gaussian distribution: $N([\omega_i^t]) \sim \mathcal{N}(0, \sigma^2_{\text{agg}})$. The remainder of the aggregation entails the scaling of freshly[9] encrypted updates with noise mean $\mu = 0$ by the computed weights. Ciphertext/ciphertext multiplication induces a Gaussian noise with mean $\mu_1\mu_2$ where μ_1 and μ_2 are respectively first and second ciphertexts noise mean. Consequently, scaling the updates by the weights ensures the preservation of a Gaussian noise with a mean of 0. Summing ciphertexts with zero-mean noise distributions results in a zero-mean Gaussian noise. Hence, the mean μ_{agg} of the noise polynomial e_{agg} is equal to 0.
- Variance: The evaluation of the variance involves three main operations: (1) subtracting $[F_i^t]_{\mathsf{FHE.pk}}$ from F_g, which does not impact the noise variance of the ciphertext $[F_i^t - F_g]_{\mathsf{FHE.pk}}$, (2) evaluating the polynomial $-\beta x^2 + 1$ on ($[F_i^t - F_g^t]_{\mathsf{FHE.pk}}$) follows the ciphertext/ciphertext multiplication noise-growth, that is, $\beta^2 2N\sigma^2_{\text{fresh}} + 4\sigma^4_{\text{fresh}}\|F_i^t - F_g^t\|_2$. And (3) encrypting scalar values implies encoding them in each of the ($\frac{N}{2}$) slots of the encoded polynomial, that is $\|F_i^t - F_g^t\|_2 = \sqrt{\frac{N}{2}}|F_i^t - F_g^t|$. The sum of the n independently scaled updates by their respective weights results in an additive noise. Furthermore, the summed ciphertexts (scaled updates) all have equal noise variance, since they originate from the same independent homomorphic encryption followed by identical FHE computation. Thus, the resulting variance (σ^2_{agg}) is the sum of the identical variances.

□

6.2 Collusion-Resistance through CKKS

This section presents a security analysis of the proposed solution w.r.t. the considered threat model in Section 4. The proposed solution relies on a t-out-of-n threshold cryptographic scheme, ensuring that any subset of $t - 1$ partial decryptions does not reveal any additional information about the associated plaintexts.
Considering that $t = B_{\text{passive}} + 1$, we hereafter discuss the security of the proposed solution against different adversaries.

[9]A ciphertext that hasn't been homomorphically evaluated.

6.2.1 Honest-but-curious clients. Our framework employs a two-pronged approach to defend against honest-but-curious adversaries:

(1) threshold decryption: the use of a (t, n) threshold decryption scheme ensures that at least t participating clients, including at least one honest client, must provide their partial decryptions to reconstruct the final model parameters. This requirement, with $t > B_{\text{passive}}$ guarantees that even if B_{passive} honest-but-curious clients collude, they cannot decrypt the model parameters without the participation of an additional honest client.
(2) DP protection: all clients are provided with a noisy global model generated from the collaborative decryption process. This noisy model offers a level of (ϵ, δ)-DP w.r.t. the CKKS parameter set and the number of clients (n), as detailed in Section 6.1.

6.2.2 Honest-but-curious server. Clients' updates are safeguarded from an honest-but-curious server through the FHE layer of the CKKS algorithm, which enables the server to obliviously process both fairness metrics evaluations and clients' updates, preventing the server from gaining access to the raw update data.

6.2.3 Colluding entities. In a collaborative learning framework where at least t clients are required for decryption, two main collusion scenarios arise. First, a client-to-client collusion occurs when colluding clients leverage their respective knowledge to access their peers' model updates. The bound on the number of potentially honest-but-curious actors ensures that colluding entities cannot extract further information from the encrypted traffic, adhering to the security assumptions of threshold schemes. Specifically, any set of $t - 1$ partial decryptions reveals no additional information about the underlying plaintext. Second, a client-server collusion involves a client sharing the plaintext aggregated model with the server after successful decryption. In this case, the Gaussian mechanism employed for approximate fairness-aware computation ensures that the honest-but-curious server can only access a noisy aggregated model from potential colluding clients, significantly limiting its ability to infer sensitive information.

7 Experimental Results

Experimental setup. The framework consists of two modules: (1) an FHE module for homomorphic aggregation, collaborative decryption, and threshold scheme setup, and (2) a learning module for clients' local training tasks. The FHE module uses the Lattigo library[10], which implements CKKS and a t-out-of-N variant. Performance experiments were run on a 12th Gen Intel Core i7-12700H with 22G memory, while the full FL workflow (Table 8) was executed on an HPC cluster using 10 nodes, each with 2 Tesla P100 GPUs.

FHE implementation analysis. The first batch of experiments focuses on the runtime of different homomorphic operations, including the framework's setup phase, in which clients collaboratively bootstrap a multi-key CKKS scheme, and the homomorphic execution of the fairness-aware aggregation. We perform several setups for different values of t and n, following two distinct situations: (1) the majority of clients are honest, and therefore decryption threshold is set to a small proportion of clients, and (2) the majority of

[10]https://github.com/tuneinsight/lattigo

clients are semi-honest, hence, decryption requires a larger proportion of participants [11]. We vary the size (secret polynomial S degree) of the secret key $log_2(N)$, which also directly constraints floating point vectors that can be encrypted in a single ciphertext to $log_2(N)-1$. These values are chosen to ensure that at least weights, and biases of a dense layer for an appropriate classifier for *Adult*, and *Compas* fit within a single CKKS ciphertext, and that at least every flattened weight vector of the convolution layers for FairFace fits a single CKKS ciphertext as well. The runtime results of these different settings are reported in Tables 4 and 5 for the setup phase, while Table 6 presents the FHE aggregation runtime.

Table 4: Sequential setup time in seconds (s) for the honest minority case ($\frac{3n}{4} \leq t \leq \frac{5n}{4}$).

$log_2(N)$	Participants (n, t) : Honest but curious majority				
	(100, 60)	(100, 65)	(100, 70)	(100, 75)	(100, 80)
11	23	23	26	27	29
12	47	53	58	62	68
13	102	111	117	126	146
14	199	205	240	232	330
15	389	395	416	532	676

Table 5: Sequential setup time in seconds (s) for the honest majority case ($\frac{n}{10} \leq t \leq \frac{n}{5}$).

$log_2(N)$	Participants (n, t) : Honest majority				
	(100, 5)	(100, 10)	(100, 15)	(100, 20)	(100, 25)
11	3	4	6	8	9
12	4	8	12	15	19
13	9	16	24	32	41
14	19	35	53	63	79
15	18	73	103	125	164

Table 6: Aggregation time in seconds for *Adult*, and *Compas*.

$log_2(N)$	Participants n.				
	50	100	200	300	500
11	1	2	4	7	11
12	2	5	9	14	23
13	5	10	20	29	47
14	10	19	39	59	100
15	21	42	89	142	195

Bandwidth consumption and communication overhead. Table 7 reports the bandwidth usage (*i.e.*, data exchanged between clients at the collaborative decryption) as well as the communication overhead with respect to a common network setting and low latency equals to $5ms$ and a common bandwidth that is around $1Gbps$. For our analysis, we considered various network settings, with the results of our experiments being reported in Appendix I.

[11] honest minority imposes a larger degree of the Shamir polynomial that produces the set public shares, therefore, strongly impacting performance.

- *Setup* — The construction of Mouchet *et al.* [67] involves all parties broadcasting shares of their additive secret key, as an evaluation of a secret, bi-variate polynomial on each of the remaining $n-1$ other parties' Shamir's public points. The RLWE nature of the secret implies that the shares and public points are degree $N-1$ polynomials with coefficients in $\mathbb{Z}_Q$. Therefore, the setup communications involve two broadcast rounds: one for the Shamir public points, and a second one for the polynomial evaluations on each of these Shamir public points. In total, $2n(n-1)N\log_2(Q)$ bits of data are exchanged between clients during setup. More details are provided in Appendix E.2.
- *Decryption* — During the collaborative decryption phase, clients generate t-out-t additive keys by (at least) t active parties, from their respective n shares of the initial n additive secret keys. Apart from revealing the active parties, the process of computing the t (a.k.a. thresholdization) does not require data exchanged other than what has been already exchanged during the setup. The last step consists of broadcasting the respective partial decryption by each of the t active parties. A partially decrypted message consists of a degree N polynomial with coefficients in $\mathbb{Z}_Q$. Thus, the size of data exchanged (for minimal t active parties) is $t(t-1)N\log_2(Q)$ bits for a single ciphertext. Encrypting a model update requires $\lceil \frac{dim(\theta_i^t)}{N/2} \rceil$ ciphertexts which is, 3 to 6 ciphertexts for *Adult*, and Compas classifiers, and from 50 to 200 ciphertexts for FairFace classifiers. For instance, FairFace classifiers have around 2 million parameters, so with 2^{14} coefficient per ciphertexts, results in 120 ciphertexts for the full model.

Table 7: Bandwidth consumption (GB) and communication overhead (in seconds) for collaborative decryption of a single ciphertext encrypting $\frac{N}{2}$ model parameters, $\log_2(Q) = 60$.

$log_2(N)$	Performances	Active parties t				
		3	5	10	15	20
11	Bandwidth (GB)	0.09	0.30	1.38	3.22	5.83
	Overhead (s)	0.01	0.03	0.14	0.32	0.58
12	Bandwidth (GB)	0.18	0.61	2.76	6.45	11.67
	Overhead (s)	0.03	0.07	0.30	0.68	1.17
13	Bandwidth (GB)	0.37	1.23	5.53	12.90	23.35
	Overhead (s)	0.06	0.14	0.64	1.48	2.58
14	Bandwidth (GB)	0.74	2.46	11.06	25.80	46.69
	Overhead (s)	0.12	0.28	1.27	2.96	5.17
15	Bandwidth (GB)	1.47	4.91	22.11	51.60	93.34
	Overhead (s)	0.23	0.54	2.40	5.60	9.90

Datasets partitioning for heterogeneous distributions. The framework is applied to three datasets: *Adult*, *Compas* and FairFace. We use the Dirichlet sampling method that samples distributions from the probability *simplex* (a distribution over the distribution space) to divide the datasets into n distinct datasets. This method allows the monitoring of the homogeneity or heterogeneity of the n distributions of the selected sensitive attribute[12] through

[12] This method does not allow full monitoring of the fairness of the updates, it simply allows to control the under/over-representation of a group in the dataset, which often has a small impact on fairness.

a concentration parameter [13] $\alpha \in \mathbf{R}^k$ where k is the number of classes of the selected attribute. For both *Adult-Census-Income*, and FairFace datasets, we choose the 'Race' attribute for the Dirichlet sampling, and measure fairness (EOD/SPD) for the Black/White groups. Similarly, for Compas dataset, we set the 'Female' binary attribute as the attribute of interest for sampling, and measure fairness using the same metrics. The experiments are conducted with $\beta = 1$. Table 8 depicts the results obtained after 300 learning iterations on the three datasets, with $n = 10$ clients respectively using FairBatch [80] and Reweighing [54] strategies to locally preprocess datasets, and enhance the fairness of the model updates throughout iterations. For CKKS, parameters, are selected such that the degree of the RLWE polynomials allows to encrypt at least one layer's parameters ($log_2(N) = 13$), and noise parameters are chosen to minimize approximation errors ($log_2(\Delta) = 65$), and ciphertext modulus Q that enables two multiplications, while keeping standard security levels $\lambda = 128$ [19].

Table 8: Encrypted `FairFed` performance global model performance after 300 learning iterations.

Measures		Heterogeneity level α				
		0.1	0.5	1	10	50
Adult	Acc	0.8059	0.7939	0.7949	0.8015	0.7959
	Precision	0.8812	0.8560	0.8215	0.8610	0.8327
	Recall	0.6612	0.7631	0.7189	0.7134	0.7377
	EOD	-0.02410	0.0054	0.0492	0.01497	0.0137
	SPD	-0.00993	-0.126	-0.0970	-0.1103	-0.1007
Compas	Acc	0.6608	0.65257	0.6511	0.6529	0.6482
	Precision	0.7531	0.7147	0.6929	0.6493	0.6353
	Recall	0.5901	0.6100	0.6353	0.5998	0.6905
	EOD	-0.053	-0.0552	-0.0703	-0.0497	-0.0353
	SPD	-0.109	-0.1285	-0.1165	-0.1680	-0.1195
FairFace	Acc	0.9250	0.9343	0.9388	0.9396	0.9413
	Precision	0.8791	0.0.8904	0.8871	0.9117	0.9083
	Recall	0.9730	0.9588	0.9603	0.9621	0.9801
	EOD	0.0202	0.0178	0.0152	0.0098	0.0109
	SPD	0.0137	0.01401	0.0081	0.0075	0.0071

Discussion of the obtained results.

- *FHE Overhead*— The FHE layer brings a natural computational overhead to the fairness-aware aggregation. However, the homomorphic circuit of our solution is limited to a small number of linear operations (degree 2 polynomial evaluation and scaling models). Alternatively, the key setup depends on the proportion of dishonest clients. A large proportion of honest-but-curious clients enforces the use of larger degree Shamir polynomials, to require a higher number of shares for full decryption, impacting the local computation time of the shares (c.f., App. E.2). Nevertheless, the FL protocol involves local learning, which is the most expensive operation, making the FHE overhead only induce an extra 5% of the entire protocol runtime for *Adult* and Compas training and only 2% for FairFace training, due to significantly larger classifiers (CNNs), and dataset, making the local training phase more computationally demanding.
- *Predictive Performance and Fairness*- Comparing the measures in [30] on *Adult* and Compas datasets, with our results show a slight degradation in the final aggregated model performance. This is primarily due to the approximations introduced by the FHE layer during the fairness-aware aggregation. Nevertheless, the impact on fairness remains limited, thus preserving the effectiveness of the `FairFed` method when transferred to the homomorphic domain. The noise introduced by the approximations in the FHE fairness-aware aggregation, however, has the potential to enhance privacy, behaving similarly to a Gaussian mechanism. Regarding the FairFace dataset, we compare our results to the ones obtained in [50], in which centralized learning was performed to assess the utility of the data. Similar to *Adult* and Compas, the accuracy degradation from FHE approximations is also observed for FairFace, as the centralized learning of [50] yields a classifier of 0.95 accuracy. As for fairness evaluation, unlike *Adult*, and Compas datasets that have inherent bias, FairFace was initially created to eliminate racial bias in facial recognition systems [50]. Hence, the induced classifiers already have low unfairness values for "Race". Baseline values for a centralized training classifier are 0.0314, and 0.0203 for EOD and SPD respectively. Comparatively, the FL-based classifiers with homomorphic fairness-aware aggregation show a slight improvement in these values, especially for low heterogeneity ($\alpha = \{10, 50\}$).

Experimental evaluation of e_{agg}. Interactions between the distribution of e_{agg}, and elements of the collaborative learning are measured by the distance between an exact fairness-aware aggregation and an approximate one from encrypted fairness measures and updates. We perform several aggregations with various parameters of the CKKS scheme that have a significant impact on the precision loss, *e.g.*, the scaling factor Δ, σ^2_{init}, and the number of updates (n). The distributions of 1000 approximation errors from different locations in the aggregated model layers are represented in figure 7. The shape of the distribution followed by the coefficients[14] of e_{agg} confirms the Gaussian mechanism analysis in section 6.1. The impact of Δ on the distribution is significant. It enhances the precision of the FHE computations at the rounding operations in the encoding/decoding and rescaling steps, where a substantial part of the precision loss occurs, preventing the loss of least significant bits of the RLWE coefficients. Nevertheless, its impact is inversely proportional to the privacy level. Indeed, higher Δ's reduce the variance of the induced Gaussian mechanism from homomorphic approximations, leading to lower privacy levels (higher ϵ values). We estimate ϵ by setting δ (failure probability) to multiple values $\{e^{-4}, e^{-6}, e^{-7}\}$ and the experimentally observed σ^2_{agg} from the set of CKKS parameters with RLWE error polynomial coefficients sampled from a zero-mean Gaussian distribution with standard deviation $\sigma_{\text{init}} = 3.2$ [2]. Finally, n is set to $\{10, 50, 100\}$.

Bounding the sensitivity. One last crucial step to formally estimate a privacy budget is to compute a bound to the sensitivity of the fairness-aware aggregation when viewed as a randomized mechanism. This mechanism operates on the union of clients' datasets $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$. Changing a single entry in $\mathcal{D}$ implies a change in

[13] This parameter defines the probability density function on the probability *simplex* $\{(x_0, \cdots, x_{k-1}) \in \mathbf{R}^k : x_i \geq 0 \text{ and } \sum_{i=0}^{k-1} x_i = 1\}$

[14] The distribution of the RLWE error polynomial (Subtraction of plaintext aggregation from the decrypted one after FHE aggregation.

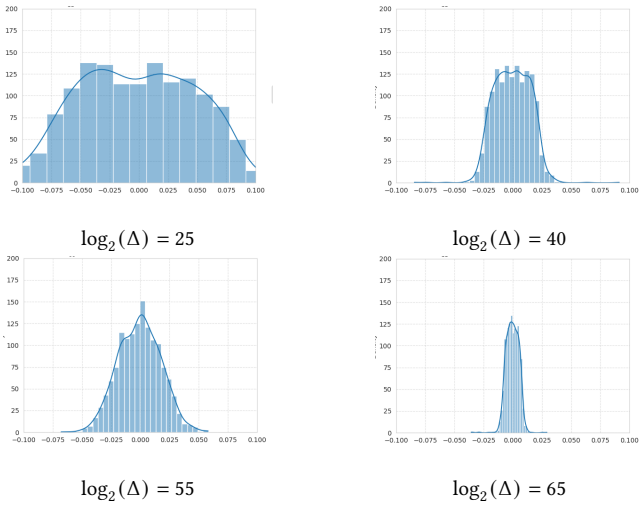


Figure 7: Distribution of e_{agg} for different sizes of Δ and $n = 3$.

a single dataset ($\mathcal{D}_j$). Hence, this results in sensitivity only at the client's j update θ_j^t, leaving the $n-1$ other updates consistent. Thus, sensitivity originates from local training (SGD).

To bound the sensitivity of local training, we rely on l_2 gradients clipping as described in [1]. Each client i scales its per-entry gradient vectors $g_i(x)$ within the ball of radius C, whenever their norm exceeds the threshold C, at every local training epoch. That is:

$$g_i(x) = g_i(x) \cdot \min\left(1, \frac{C}{\|g_i(x)\|_2}\right)$$

Under this assumption, a bound on the sensitivity of local SGD but also the fairness-aware aggregation's sensitivity is C. Hence, from theorem 2.3 [28], a formal estimate of the privacy budget ϵ given CKKS parameters, number of clients and clipping norm C is:

$$\epsilon = \frac{\sqrt{2\log(\frac{1.25}{\delta})}C}{\sigma_{\text{agg}}}$$

Table 9 reports the estimations of ϵ for fixed CKKS parameters, the number of clients n for a model trained with FL on *Adult*, while Tables 11 and 12 in Appendix E report the estimations for Compas, and FairFace, respectively[15].

Imperfect Gaussian noise. The Gaussian nature of the approximation noise from the homomorphic fairness-aware aggregation is consistent with the definition of the Gaussian mechanism. Furthermore, the non-perfect aspect of this noise, due to its discrete nature and approximations, has been investigated in [13], showing both experimentally and theoretically, similar privacy guarantees to the continuous and perfect additive Gaussian noise (as in [28]). Lastly, the modular (coefficient-wise) nature of RLWE ciphertexts makes the approximation error of CKKS follow a Gaussian distribution over $\mathbf{Z}_Q$ (discrete and modular). However, Q is usually huge next to standard deviations σ_{init}, and even σ_{agg} (e.g. $\sigma_{\text{init}} = 3.2$, and $Q = 2^{60}$). Therefore, the likelihood of noise samples exceeding Q (*wrapping-up*) is vanishingly small [16]. As a result, the modular nature of the distribution of approximation errors can be largely disregarded when considering DP guarantees.

[15] As pointed out in [1], a good practice is to set C to be the median norm of the unclipped gradients throughout training. We empirically estimate this value for *Adult*, Compas and FairFace training to be respectively at most 0.5, 0.66 and 0.82, following uniform initialization in [-1, 1].

Table 9: Estimation of ϵ from approximate homomorphic fairness-aware aggregation for *Adult*.

δ	$log_2(N)$	n	$log_2(\Delta)$	σ^2_{agg}	ϵ
e^{-4}	13	10	40	$2.992 \cdot 10^{-3}$	36.10
	14	50	40	$3.840 \cdot 10^{-2}$	9.46
	15	100	40	$8.293 \cdot 10^{-4}$	70.36
e^{-6}	13	10	50	$3.2335 \cdot 10^{-3}$	≥ 100
	14	50	50	$9.6135 \cdot 10^{-2}$	10.04
	15	100	50	$1.855 \cdot 10^{-2}$	23.00
e^{-7}	13	10	40	$3.503 \cdot 10^{-1}$	6.53
	14	50	40	$4.235 \cdot 10^{-1}$	5.91
	15	100	40	$9.32 \cdot 10^{-1}$	40.28

Differentially-Private aggregation via CKKS parameters. Tables 9, 11, 12 and Lemma 6.1 demonstrate that for a given CKKS parameters, and FL setup (n, training hyper-parameters, and aggregation strategy), one can estimate the variance of the Gaussian noise induced from a CKKS homomorphic aggregation when this scheme is employed as a secure aggregation mechanism. Essentially, the CKKS scheme, under this FL training setup, can be parameterized for a dual purpose: (1) achieve a proper cryptographic security level $\lambda \geq 128$ bits, (2) ensure a sufficient noise variance σ^2_{agg} at each aggregation to achieve a desired privacy budget ϵ. Hence, achieving end-to-end privacy, throughout the FL process.

8 Conclusion

In this work, we demonstrate the importance of carefully evaluating the privacy leakage in collaborative training protocols that aim at enhancing group fairness and require the disclosure of different fairness-related measures from participating entities. Indeed, most privacy attacks rely on the assumption of overfitting models, which expresses the disparities in the model's performance when evaluated on training and test data and is conceptually very close to the idea of group fairness metrics, which also measures and expresses a difference in a model's behaviour when evaluated on two distinct groups, defined by a sensitive attribute. Thus, sharing it along with a model's parameters or predictions should be meticulously considered from a privacy standpoint. Other leakages are shown to come from weight disparities as most participating clients will experience a higher leakage. Beyond, emphasizing these privacy risks, we proposed a framework that conciliates fairness and privacy with small computational and communication overheads. Finally, we leverage CKKS's approximations to provide differentially-private and fair aggregation.

[16] Using Chebyshev inequality, this probability is bounded by $\frac{\sigma^2_{\text{init}}}{Q^2} \approx 2^{-100}$.

Acknowledgments

This work was supported by the France ANR project ANR-22-CE39-0002 EQUIHID and by the France 2030 ANR Projects ANR-22-PECY-003 SecureCompute and ANR-23-PECL-0009 TRUSTINCloudS.

Sébastien Gambs is supported by the Canada Research Chair program, a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) as well as the DEEL Project CRDPJ 537462-18 funded by the NSERC and the Consortium for Research and Innovation in Aerospace in Québec (CRIAQ).

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16, Vol. N/A)*. ACM, New York, NY, USA, N/A. https://doi.org/10.1145/2976749.2978318

[2] 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. 2019. Homomorphic Encryption Standard. Cryptology ePrint Archive, Paper 2019/939. https://eprint.iacr.org/2019/939 https://eprint.iacr.org/2019/939.

[3] Héber H. Arcolezi, Karima Makhlouf, and Catuscia Palamidessi. 2023. *(Local) Differential Privacy has NO Disparate Impact on Fairness.* Springer Nature Switzerland, Switzerland, 3–21. https://doi.org/10.1007/978-3-031-37586-6_1

[4] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Tobias Hepp, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. 2020. MedGAN: Medical image translation using GANs. *Computerized Medical Imaging and Graphics* 79 (jan 2020), 101684. https://doi.org/10.1016/j.compmedimag.2019.101684

[5] Gilad Asharov, Abhishek Jain, and Daniel Wichs. 2011. Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE. Cryptology ePrint Archive, Paper 2011/613. https://eprint.iacr.org/2011/613 https://eprint.iacr.org/2011/613.

[6] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. 2019. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems* 32, N/A (2019), N/A.

[7] James Bell, K. A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. 2020. Secure Single-Server Aggregation with (Poly)Logarithmic Overhead. Cryptology ePrint Archive, Paper 2020/704. https://doi.org/10.1145/3372297.3417885 https://eprint.iacr.org/2020/704.

[8] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy Preserving Machine Learning. Cryptology ePrint Archive, Paper 2017/281. https://eprint.iacr.org/2017/281 https://eprint.iacr.org/2017/281.

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy Preserving Machine Learning. Cryptology ePrint Archive, Paper 2017/281. https://eprint.iacr.org/2017/281 https://eprint.iacr.org/2017/281.

[10] Christina Boura, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev. 2018. CHIMERA: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes. Cryptology ePrint Archive, Paper 2018/758. https://eprint.iacr.org/2018/758 https://eprint.iacr.org/2018/758.

[11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2011. Fully Homomorphic Encryption without Bootstrapping. Cryptology ePrint Archive, Paper 2011/277. https://eprint.iacr.org/2011/277 https://eprint.iacr.org/2011/277.

[12] David Byrd and Antigoni Polychroniadou. 2020. Differentially Private Secure Multi-Party Computation for Federated Learning in Financial Applications. arXiv:2010.05867 [cs.CR]

[13] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. 2021. The Discrete Gaussian for Differential Privacy. arXiv:2004.00010 [cs.DS]

[14] Hongyan Chang and Reza Shokri. 2021. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, IEEE, Piscataway, NJ, USA, 292–303.

[15] Hongyan Chang and Reza Shokri. 2023. Bias Propagation in Federated Learning. arXiv:2309.02160 [cs.LG]

[16] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially Private Empirical Risk Minimization. arXiv:0912.0071 [cs.LG]

[17] Huiqiang Chen, Tianqing Zhu, Tao Zhang, Wanlei Zhou, and Philip S Yu. 2023. Privacy and Fairness in Federated Learning: on the Perspective of Trade-off. *Comput. Surveys* N/A, N/A (2023), N/A.

[18] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*, Tsuyoshi Takagi and Thomas Peyrin (Eds.). Springer International Publishing, Cham, 409–437.

[19] Jung Hee Cheon, Yongha Son, and Donggeon Yhee. 2021. Practical FHE parameters against lattice attacks. Cryptology ePrint Archive, Paper 2021/039. https://eprint.iacr.org/2021/039 https://eprint.iacr.org/2021/039.

[20] Diego Chialva and Ann Dooms. 2018. Conditionals in Homomorphic Encryption and Machine Learning Applications. Cryptology ePrint Archive, Paper 2018/1032. https://eprint.iacr.org/2018/1032 https://eprint.iacr.org/2018/1032.

[21] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. 2018. TFHE: Fast Fully Homomorphic Encryption over the Torus. Cryptology ePrint Archive, Paper 2018/421. https://eprint.iacr.org/2018/421 https://eprint.iacr.org/2018/421.

[22] Alexandra Chouldechova. 2016. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. arXiv:1610.07524 [stat.AP]

[23] Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. 2022. On the precision loss in approximate homomorphic encryption. Cryptology ePrint Archive, Paper 2022/162. https://eprint.iacr.org/2022/162 https://eprint.iacr.org/2022/162.

[24] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. 2019. On the Compatibility of Privacy and Fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization* (Larnaca, Cyprus) *(UMAP'19 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 309–315. https://doi.org/10.1145/3314183.3323847

[25] Emiliano De Cristofaro. 2020. An overview of privacy in machine learning. *arXiv preprint arXiv:2005.08679* N/A, N/A (2020), N/A.

[26] Emiliano De Cristofaro. 2021. A critical overview of privacy in machine learning. *IEEE Security & Privacy* 19, 4 (2021), 19–27.

[27] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006 (Lecture Notes in Computer Science, Vol. 3876)*. Springer, Tiergartenstrasse 17, 69121 Heidelberg, Germany, 265–284. https://doi.org/10.1007/11681878_14

[28] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.

[29] Ahmed Roushdy Elkordy, Jiang Zhang, Yahya H. Ezzeldin, Konstantinos Psounis, and Salman Avestimehr. 2022. How Much Privacy Does Federated Learning with Secure Aggregation Guarantee? arXiv:2208.02304 [cs.LG]

[30] Yahya H. Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and Salman Avestimehr. 2023. FairFed: enabling group fairness in federated learning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'23/IAAI'23/EAAI'23)*. AAAI Press, Palo Alto, CA, USA, Article 842, 9 pages. https://doi.org/10.1609/aaai.v37i6.25911

[31] Junfeng Fan and Frederik Vercauteren. 2012. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, Paper 2012/144. https://eprint.iacr.org/2012/144 https://eprint.iacr.org/2012/144.

[32] Zhenan Fan, Huang Fang, Zirui Zhou, Jian Pei, Michael P Friedlander, Changxin Liu, and Yong Zhang. 2022. Improving fairness for data valuation in horizontal federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, IEEE, Piscataway, NJ, USA, 2440–2453.

[33] Michael Feldman, Sorelle Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. arXiv:1412.3756 [stat.ML]

[34] Julien Ferry, Ulrich Aïvodji, Sébastien Gambs, Marie-José Huguet, and Mohamed Siala. 2022. Exploiting Fairness to Enhance Sensitive Attributes Reconstruction. arXiv:2209.01215 [cs.LG]

[35] Xiuting Gu, Tianqing Zhu, Jie Li, Tao Zhang, and Wei Ren. 2020. The Impact of Differential Privacy on Model Fairness in Federated Learning. In *Network and System Security*, Mirosław Kutyłowski, Jun Zhang, and Chao Chen (Eds.). Springer International Publishing, Cham, 419–430.

[36] Kyoohyung Han and Dohyeong Ki. 2019. Better Bootstrapping for Approximate Homomorphic Encryption. Cryptology ePrint Archive, Paper 2019/688. https://eprint.iacr.org/2019/688 https://eprint.iacr.org/2019/688.

[37] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. arXiv:1610.02413 [cs.LG]

[38] Yuzi He, Keith Burghardt, and Kristina Lerman. 2020. A Geometric Solution to Fair Representations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (New York, NY, USA) *(AIES '20)*. Association for Computing Machinery, New York, NY, USA, 279–285. https://doi.org/10.1145/3375627.3375864

[39] Wei Huang, Tianrui Li, Dexian Wang, Shengdong Du, Junbo Zhang, and Tianqiang Huang. 2022. Fairness and accuracy in horizontal federated learning. *Information Sciences* 589 (2022), 170–185.

[40] Muhammad Akbar Husnoo, Adnan Anwar, Nasser Hosseinzadeh, Shama Naz Islam, Abdun Naser Mahmood, and Robin Doss. 2023. A Secure Federated Learning Framework for Residential Short Term Load Forecasting. arXiv:2209.14547 [cs.CR] https://arxiv.org/abs/2209.14547

[41] Ilia Iliashenko and Vincent Zucca. 2021. Faster homomorphic comparison operations for BGV and BFV. Cryptology ePrint Archive, Paper 2021/315. https://eprint.iacr.org/2021/315 https://eprint.iacr.org/2021/315.

[42] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. 2019. Differentially private fair learning. In *International Conference on Machine Learning*. PMLR, PMLR, 1500 E. Eugene Street, Suite 200, Eugene, OR 97401, USA, 3000–3008.

[43] Tayyebeh Jahani-Nezhad, Mohammad Ali Maddah-Ali, Songze Li, and Giuseppe Caire. 2022. SwiftAgg: Communication-Efficient and Dropout-Resistant Secure Aggregation for Federated Learning with Worst-Case Security Guarantees. arXiv:2202.04169 [cs.IT] https://arxiv.org/abs/2202.04169

[44] Peter Kairouz, Ziyu Liu, and Thomas Steinke. 2022. The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation. arXiv:2102.06387 [cs.LG] https://arxiv.org/abs/2102.06387

[45] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.

[46] Yilin Kang, Yong Liu, Ben Niu, Xinyi Tong, Likun Zhang, and Weiping Wang. 2020. Input Perturbation: A New Paradigm between Central and Local Differential Privacy. arXiv:2002.08570 [cs.LG]

[47] Yilin Kang, Yong Liu, Ben Niu, Xinyi Tong, Likun Zhang, and Weiping Wang. 2020. Input Perturbation: A New Paradigm between Central and Local Differential Privacy. arXiv:2002.08570 [cs.LG]

[48] Tanveer Khan and Antonis Michalas. 2023. Learning in the Dark: Privacy-Preserving Machine Learning using Function Approximation. arXiv:2309.08190 [cs.CR] https://arxiv.org/abs/2309.08190

[49] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. 2021. Disparate Vulnerability to Membership Inference Attacks. arXiv:1906.00389 [cs.LG] https://arxiv.org/abs/1906.00389

[50] Kimmo Kärkkäinen and Jungseock Joo. 2019. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age. arXiv:1908.04913 [cs.CV]

[51] Eunsang Lee, Joon-Woo Lee, Jong-Seon No, and Young-Sik Kim. 2020. Minimax Approximation of Sign Function by Composite Polynomial for Homomorphic Comparison. Cryptology ePrint Archive, Paper 2020/834. https://eprint.iacr.org/2020/834 https://eprint.iacr.org/2020/834.

[52] Eunsang Lee, Joon-Woo Lee, Jong-Seon No, and Young-Sik Kim. 2022. Minimax Approximation of Sign Function by Composite Polynomial for Homomorphic Comparison. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2022), 3711–3727. https://doi.org/10.1109/TDSC.2021.3105111

[53] Joon-Woo Lee, Hyungchul Kang, Yongwoo Lee, Woosuk Choi, Jieun Eom, Maxim Deryabin, Eunsang Lee, Junghyun Lee, Donghoon Yoo, Young-Sik Kim, and Jong-Seon No. 2022. Privacy-Preserving Machine Learning With Fully Homomorphic Encryption for Deep Neural Network. *IEEE Access* 10 (2022), 30039–30054. https://doi.org/10.1109/ACCESS.2022.3159694

[54] Peizhao Li and Hongfu Liu. 2022. Achieving Fairness at No Utility Cost via Data Reweighing with Influence. arXiv:2202.00787 [cs.LG]

[55] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–36.

[56] Suyun Liu and Luis Nunes Vicente. 2022. Accuracy and Fairness Trade-offs in Machine Learning: A Stochastic Multi-Objective Approach. arXiv:2008.01132 [cs.LG]

[57] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2021. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. arXiv:2102.02551 [cs.CR] https://arxiv.org/abs/2102.02551

[58] Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao. 2022. Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data* N/A, N/A (2022), N/A.

[59] Zeyu Liu, Daniele Micciancio, and Yuriy Polyakov. 2021. Large-Precision Homomorphic Sign Evaluation using FHEW/TFHE Bootstrapping. Cryptology ePrint Archive, Paper 2021/1337. https://eprint.iacr.org/2021/1337 https://eprint.iacr.org/2021/1337.

[60] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. 2013. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)* 60, 6 (2013), 1–35.

[61] Yiping Ma, Jess Woods, Sebastian Angel, Antigoni Polychroniadou, and Tal Rabin. 2023. Flamingo: Multi-Round Single-Server Secure Aggregation with Applications to Private Federated Learning. Cryptology ePrint Archive, Paper 2023/486. https://eprint.iacr.org/2023/486 https://eprint.iacr.org/2023/486.

[62] Abbass Madi, Oana Stan, Aurélien Mayoue, Arnaud Grivet-Sébert, Cédric Gouy-Pailler, and Renaud Sirdey. 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)*. IEEE, Piscataway, NJ, USA, 1–8. https://doi.org/10.1109/RDAAPS48126.2021.9452005

[63] Paul Mangold, Michaël Perrot, Aurélien Bellet, and Marc Tommasi. 2023. Differential Privacy has Bounded Impact on Fairness in Classification. arXiv:2210.16242 [cs.LG]

[64] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated Learning of Deep Networks using Model Averaging. *CoRR* abs/1602.05629, N/A (2016), N/A. arXiv:1602.05629 http://arxiv.org/abs/1602.05629

[65] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[66] Mansouri Mohamad, Melek Önen, Wafa Ben Jaballah, and Mauro Conti. 2023. SoK: Secure aggregation based on cryptographic schemes for federated learning. In *PETS 2023, 23rd Privacy Enhancing Technologies Symposium, 10-15 July 2023, Lausanne, Switzerland (Hybrid Conference)*, IACR (Ed.). ACM, Lausanne, N/A. IACR.

[67] Christian Mouchet, Elliott Bertrand, and Jean-Pierre Hubaux. 2022. An Efficient Threshold Access-Structure for RLWE-Based Multiparty Homomorphic Encryption. Cryptology ePrint Archive, Paper 2022/780. https://doi.org/10.1007/s00145-023-09452-8 https://eprint.iacr.org/2022/780.

[68] Christian Mouchet, Juan Troncoso-Pastoriza, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. 2020. Multiparty Homomorphic Encryption from Ring-Learning-With-Errors. Cryptology ePrint Archive, Paper 2020/304. https://doi.org/10.2478/popets-2021-0071 https://eprint.iacr.org/2020/304.

[69] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, IEEE, Piscataway, NJ, USA, 739–753.

[70] Tabitha Ogilvie. 2023. Differential Privacy for Free? Harnessing the Noise in Approximate Homomorphic Encryption. Cryptology ePrint Archive, Paper 2023/701. https://eprint.iacr.org/2023/701 https://eprint.iacr.org/2023/701.

[71] Manisha Padala, Sankarshan Damle, and Sujit Gujar. 2021. Federated Learning Meets Fairness and Differential Privacy. arXiv:2108.09932 [cs.LG]

[72] Marlotte Pannekoek and Giacomo Spigler. 2021. Investigating Trade-offs in Utility, Fairness and Differential Privacy in Neural Networks. arXiv:2102.05975 [cs.LG]

[73] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. arXiv:1802.08908 [stat.ML]

[74] Alberto Pedrouzo-Ulloa, Aymen Boudguiga, Olive Chakraborty, Renaud Sirdey, Oana Stan, and Martin Zuber. 2022. Practical Multi-Key Homomorphic Encryption for More Flexible and Efficient Secure Federated Aggregation (preliminary work). Cryptology ePrint Archive, Paper 2022/1674. https://eprint.iacr.org/2022/1674 https://eprint.iacr.org/2022/1674.

[75] John Platt and Alan Barr. 1987. Constrained Differential Optimization. In *Neural Information Processing Systems*, D. Anderson (Ed.), Vol. 0. American Institute of Physics, 1305 Walt Whitman Road, Suite 300, Melville, NY 11747-4300, USA. https://proceedings.neurips.cc/paper_files/paper/1987/file/a87ff679a2f3e71d9181a67b7542122c-Paper.pdf

[76] David Pujol, Ryan McKenna, Satya Kuppam, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. 2020. Fair decision making using privacy-protected data. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 189–199. https://doi.org/10.1145/3351095.3372872

[77] Fengyuan Qiu, Hao Yang, Lu Zhou, Chuan Ma, and LiMing Fang. 2022. Privacy Preserving Federated Learning Using CKKS Homomorphic Encryption. In *Wireless Algorithms, Systems, and Applications*, Lei Wang, Michael Segal, Jenhui Chen, and Tie Qiu (Eds.). Springer Nature Switzerland, Cham, 427–440.

[78] Maria Rigaki and Sebastian Garcia. 2023. A Survey of Privacy Attacks in Machine Learning. *Comput. Surveys* 56, 4 (Nov. 2023), 1–34. https://doi.org/10.1145/3624010

[79] Borja Rodríguez-Gálvez, Filip Granqvist, Rogier van Dalen, and Matt Seigel. 2022. Enforcing fairness in private federated learning via the modified method of differential multipliers. arXiv:2109.08604 [cs.LG]

[80] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2021. FairBatch: Batch Selection for Model Fairness. arXiv:2012.01696 [cs.LG]

[81] Timon Rückel, Johannes Sedlmeir, and Peter Hofmann. 2022. Fairness, integrity, and privacy in a scalable blockchain-based federated learning system. *Computer Networks* 202 (2022), 108621.

[82] Theo Ryffel, Edouard Dufour-Sans, Romain Gay, Francis Bach, and David Pointcheval. 2021. Partially Encrypted Machine Learning using Functional Encryption. arXiv:1905.10214 [cs.LG]

[83] Thomas Sandholm, Sayandev Mukherjee, and Bernardo A. Huberman. 2021. SAFE: Secure Aggregation with Failover and Encryption. arXiv:2108.05475 [cs.DC] https://arxiv.org/abs/2108.05475

[84] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, New Jersy, NJ, USA, 3–18.

https://doi.org/10.1109/SP.2017.41
[85] Jinhyun So, Chaoyang He, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E. Ali, Basak Guler, and Salman Avestimehr. 2022. LightSecAgg: a Lightweight and Versatile Design for Secure Aggregation in Federated Learning. arXiv:2109.14236 [cs.LG]
[86] Timothy Stevens, Christian Skalka, Christelle Vincent, John Ring, Samuel Clark, and Joseph Near. 2021. Efficient Differentially Private Secure Aggregation for Federated Learning via Hardness of Learning with Errors. arXiv:2112.06872 [cs.CR]
[87] Huan Tian, Guangsheng Zhang, Bo Liu, Tianqing Zhu, Ming Ding, and Wanlei Zhou. 2024. When Fairness Meets Privacy: Exploring Privacy Threats in Fair Binary Classifiers through Membership Inference Attacks. arXiv:2311.03865 [cs.LG] https://arxiv.org/abs/2311.03865
[88] Archit Uniyal, Rakshit Naidu, Sasikanth Kotti, Sahib Singh, Patrik Joslin Kenfack, Fatemehsadat Mireshghallah, and Andrew Trask. 2021. Dp-sgd vs pate: Which has less disparate impact on model accuracy? *arXiv preprint arXiv:2106.12576* N/A, N/A (2021), N/A.
[89] Yue Wang, Xintao Wu, and Donghui Hu. 2016. Using Randomized Response for Differential Privacy Preserving Data Collection. In *EDBT/ICDT Workshops*. EDBT/ICDT, New York, NY, USA, N/A. https://api.semanticscholar.org/CorpusID:1286991
[90] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. arXiv:1805.11202 [cs.LG]
[91] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, James Joshi, and Heiko Ludwig. 2021. FedV: Privacy-Preserving Federated Learning over Vertically Partitioned Data. arXiv:2103.03918 [cs.LG]
[92] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–36.
[93] Xubo Yue, Maher Nouiehed, and Raed Al Kontar. 2023. GIFAIR-FL: A Framework for Group and Individual Fairness in Federated Learning. *INFORMS Journal on Data Science* 2, 1 (April 2023), 10–23. https://doi.org/10.1287/ijds.2022.0022
[94] Yuchen Zeng, Hongxu Chen, and Kangwook Lee. 2021. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545* N/A, N/A (2021), N/A.
[95] Daniel Yue Zhang, Ziyi Kou, and Dong Wang. 2020. Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, IEEE, Piscataway, NJ, USA, 1051–1060.

A Table of notations

Table 10: List of notations

Acronym or Symbol	Definition
DP	Differential Privacy
EOD	Equal Opportunity Difference
FHE	Fully Homomorphic Encryption
FL	Federated Learning
GAN	Generative Adversarial Network
(R)LWE	(Ring) Learning with errors
SMC	Secure Multi-party Computation
SPD	Statistical Parity Difference
n	number of clients
$[k]$	$\{0, \cdots, k-1\}$
n_k	number of datapoints at client k
n_{sh}	number of shadow models
$\mathcal{D}_i$	Client's i local dataset
θ_i^t	Parameters update by client i at iteration t
θ_g^t	Global model parameters at iteration t
$M_\theta(x)$	Inference of model M_θ on input x
$[x]_{\mathsf{FHE.pk}}$	FHE of x with key FHE.pk
$N([x]_{\mathsf{FHE.pk}})$	Noise random variable of $[x]_{\mathsf{FHE.pk}}$

B Supervised and Federated Learning

B.1 Supervised learning

Supervised learning refers to the process of learning a set of parameters θ for a model h on a collection of data samples $(x_i)_{i\in[K]}$ where x_i is a feature vector of fixed dimension, and their corresponding labels $(y_i)_{i\in[K]}$, such that $\hat{y_i} = h_\theta(x_i)$ is as close as possible to y_i. The closeness of the predictions $\hat{y_i}$ to the labels y_i with respect to the set of parameters θ is expressed by a loss function $\mathcal{L}(y_i, \hat{y}_i, \theta)$. The training objective is to compute $\texttt{argmin}_\theta \mathcal{L}$. by iteratively minimizing the average of $\mathcal{L}$ over all samples in batches from dataset, following an SGD based approach. That is, differentiating the loss function to obtain a gradient vector $g^{(t)} = \nabla_{\theta^{(t)}} \mathcal{L}(\theta^{(t)})$, followed by parameter gradient descent $\theta^{(t+1)} \leftarrow \theta^t - \alpha g^{(t)}$ where α controls the amplitude of the descent (learning-rate).

B.2 Federated Learning

Federated Learning is a distributed learning framework wherein a model is collaboratively trained to tackle the issue of dispersed clients sharing resource without exposing raw information. Common learning strategies involve aggregating updates from data-owners. FedAvg aggregates models parameters from data-owners $\sum_{i=1}^n \frac{n_i}{\sum_{i=1}^n n_i} \theta_i^t$ to produce a global model which is again trained on clients edge. Updates can be locally computed gradients as well. Aggregated gradients result in a global gradient vector which serves to update global model with a global learning rate (η).

$$g^{t+1} = \omega_i^t g_i^t \text{ and } \theta_g^{t+1} = \theta_g^t - \eta g^{t+1}$$

Where ω_i^t scale participants updates according to a measure of its quality. *e.g.* number of local data-samples in FedAvg)

C Membership inference attacks through shadow training

Shokri, Stronati, Song and Shmatikov [84] have designed a membership inference attack that relies on the use of *shadow models*. These shadow models S_i, are trained on datasets $\mathcal{D}_i$ that closely resemble the distribution of the training dataset of h. Consequently, they exhibit close behavior to h when employed for inference tasks. The statistical correlations between the model behaviour and the membership status of the data are then captured in an attack dataset $\hat{\mathcal{D}}$ composed of tuples $(x, S_i(x), y_{member} = \{1 \text{ if } x \in \mathcal{D}_i, 0 \text{ otherwise}\})$. Finally, the attack model is trained on $\hat{\mathcal{D}}$ to capture these statistical correlations. Hence, predicting the membership label y_{member} with high accuracy when provided with the target model predictions on a given data point $(x, h(x))$.

FairGAN architecture and loss functions

In addition to a generator and the dual discriminator, FairGAN incorporates, an autoencoder model that is pre-trained, with his half decoder being incorporated to the generator in the MedGAN [4] fashion, such that $G_{Dec}(z) := Dec(G(z))$ as depicted in Figure 8. The overarching objective function in the FairGAN architecture is the sum of two functions that express the two constraints placed

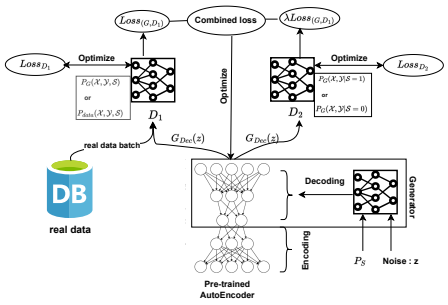


Figure 8: The FairGAN architecture.

upon the generator. G_{Dec}:

$$\min_{D_1} l(G, D_1) = E_{(x,y,s)\sim P_{\text{data}}(s)}[log(D_1(x, y, s))] + E_{\hat{s}\sim P_{\text{data}},(\hat{x},\hat{y})\sim P_G(x,y|s)}[log(1 - D_1(\hat{x}, \hat{y}, \hat{s}))]$$

It expresses the distinguishing task of D_1, which is to output 1 on real samples (from P_{data}) and 0 on fake samples (from P_G). Meanwhile, the distinguishing task of D_2 is expressed by

$$\min_{D_2} l(G, D_2) = E_{(\hat{x},\hat{y})\sim P_G(\hat{x},\hat{y}|s=1)}[log(D_1(\hat{x}, \hat{y}))] + E_{(\hat{x},\hat{y})\sim P_G(x,y|s=0)}[log(1 - D_1(\hat{x}, \hat{y}))]$$

This aims at providing an accurate prediction of s given synthetically generated label and unprotected attributes $(\hat{x}, \hat{y})$.

D Privacy budget estimation for Compas, and FairFace.

This section provides an estimation of the privacy budget ϵ from the proposed approximate homomorphic fairness-aware aggregation for both the Compas and FairFace datasets.

Table 11: Estimation of ϵ from approximate homomorphic fairness-aware aggregation for Compas.

δ	$log_2(N)$	n	$log_2(\Delta)$	σ^2_{agg}	ϵ
e^{-4}	13	10	40	$2.992 \cdot 10^{-3}$	37.90
	14	50	40	$3.840 \cdot 10^{-2}$	10.52
	15	100	40	$8.293 \cdot 10^{-4}$	71.30
e^{-6}	13	10	50	$3.2335 \cdot 10^{-3}$	≥ 100
	14	50	50	$9.6135 \cdot 10^{-2}$	9.89
	15	100	50	$1.855 \cdot 10^{-2}$	22.49
e^{-7}	13	10	40	$3.503 \cdot 10^{-1}$	6.22
	14	50	40	$4.235 \cdot 10^{-1}$	5.90
	15	100	40	$9.32 \cdot 10^{-1}$	40.30

E Cryptographic background

E.1 CKKS functionalities

- Keygeneration : The secret key polynomial s is sampled from a distribution S, and the secret key is the tuple $sk = (1, s)$. As for the public key generation, a polynomial a is uniformly sampled from $Z_q^N = \mathbb{Z}_q/(X^N + 1)$ where N is a power of two. Hence, $X^N + 1$ is cyclotomic polynomial. Another error polynomial is sampled from Z_q^N using a multivariate Gaussian distribution and the public key is $pk = ([as + e]_q, a)$.
- Relinearization – KeyGen : Performing a component-wise multiplication of two ciphertexts results in a third quadratic term in the secret key polynomial s, and results in a quadratic growth in the ciphertext size. Hence the need to transform the ciphertext back to the standard RLWE ciphertext format composed of two elements and a linear decryption in the secret key polynomial s. To do so a relinearization key is provided, which can be interpreted as an encryption of s^2 using s.
$$\texttt{RelinKey} = ([a's + e' + Ps^2]_P q, a')$$
- Encryption : Given $pk = (pk_0, pk_1)$ and $m \in Z_q^N$, sample v from the same distribution S as the secret key polynomial s, and sample e_0, e_1 from a multivariate Gaussian distribution. The encryption of m under pk is :
$$(c_0, c_1) = ([m + pk_0 v + e_0]Q_L, [p_1 v + e_2]Q_L)$$
- Decryption : Given a ciphertext $c = (c_0, c_1)$ encrypting a message $m \in Z_q^n$. Decryption procedure outputs $m' = [c_0 + c_1 \cdot s]_{q_l}$ such that m' is a good approximation of m.
- Addition : Given two ciphertexts encrypted using the same public key pk, $c_0 = (c_{0,0}, c_{0,1})$ and $c_1 = (c_{1,0}, c_{1,1})$. Homomorphic addition is performed component-wise. That is : $c_0 + c_1 = ([c_{0,0} + c_{1,0}]_{Q_l}, [c_{0,0} + c_{1,0}]_{Q_l})$
- Multiplication : Given two ciphertexts encrypted using the same public key pk, $c_0 = (c_{0,0}, c_{0,1})$ and $c_1 = (c_{1,0}, c_{1,1})$. Homomorphic multiplication (first step, also referred to as pre-multiplication) is performed component-wise. That is :
$$([c_{0,0}c_{1,0}]_{Q_l}, [c_{0,0}c_{1,1} + c_{0,1}c_{1,1}]_{Q_l}, [c_{0,1}c_{1,1}]_{Q_l})$$
- Relinearization : From the first stage of ciphertext multiplication producing a three-terms ciphertext, the goal of Relinearization is to remove the quadratic term in s, while maintaining the ability to decrypt correctly. Given a component-wise product of two ciphertexts (4) This is done by :
$$([c_{0,0}c_{1,0}]_{Q_l}, [c_{0,0}c_{1,1} + c_{0,1}c_{1,1}]_{Q_l}) + [P^{-1}[c_{0,1}c_{1,1}]_{Q_l} \cdot RelinKey]_q$$
- Rescaling : To avoid a quadratic expansion of ciphertext size through multiplications authors introduce a rescaling method that consists in multiplying both terms of the ciphertext by the

Table 12: Estimation of ϵ from approximate homomorphic fairness-aware aggregation for FairFace .

δ	$log_2(N)$	n	$log_2(\Delta)$	σ^2_{agg}	ϵ
e^{-4}	13	10	40	$2.992 \cdot 10^{-3}$	37.0
	14	50	40	$3.840 \cdot 10^{-2}$	10.21
	15	100	40	$8.293 \cdot 10^{-4}$	70.1
e^{-6}	13	10	50	$3.2335 \cdot 10^{-3}$	≥ 100
	14	50	50	$9.6135 \cdot 10^{-2}$	10.42
	15	100	50	$1.855 \cdot 10^{-2}$	23.1
e^{-7}	13	10	40	$3.503 \cdot 10^{-1}$	6.33
	14	50	40	$4.235 \cdot 10^{-1}$	5.51
	15	100	40	$9.32 \cdot 10^{-1}$	40.32

scaling factor Δ followed by a coefficient-wise rounding to the nearest integer: $(\lfloor \frac{1}{\Delta} c_0 \rceil, \lfloor \frac{1}{\Delta} c_1 \rceil)$.

Therefore a full multiplication in CKKS consists in a component-wise multiplication followed by a relinearization and a rescaling.

Table 13 gives the formulas for ciphertext noise growth with respect to CKKS operations.

E.2 Threshold multi-key RLWE-based schemes

Threshold cryptography refers to encryption schemes with collaborative decryption mechanisms. That is, each participant holds a share sk_i of the (global) secret key sk, which grants him the ability to perform a partial decryption of a ciphertext $\hat{m}_i = \texttt{Partial\_Dec}(sk_i, c)$. An access structure $S \subset \texttt{PowerSet}(P)$ is the collection of all subsets of parties with the ability to recover the plaintext message using their secret key shares. A t-out-of-n scheme refers to the access structure containing subsets of size t.

$\forall P \subset S$ we have $m = \texttt{Dec}(sk, c) = \texttt{Full\_Dec}(P_{Dec})$

with $P_{Dec} = \{\hat{m}_i \ \forall i \text{ with } p_i \in P\}$, the set of partial decryptions from parties in P.

The fundamental security assumption of threshold schemes is that any strict subset of $P \in S$ provides no additional knowledge regarding m beyond what is already provided by the ciphertext c.

Mouchet *et al.* [68] introduced an additive [17] n-out-of-n multiparty construction for ring learning with errors structure, and therefore compatible with all RLWE-based homomorphic schemes. Namely, BGV [11], BFV [31] and CKKS [18]. It was later extended to t-out-of-n access structures for any $t \leq n$ using the *share-resharing* technique by Asharov *et al.* [5] that enables to move from an n-out-of-n construction to a t-out-of-t one, by *re-sharing* the additive locally generated secret key through Shamir secret sharing, and taking advantage of the commutative nature between additive decryption reconstruction, and the interpolation of the secret key. That is,

$$s = \sum_{i=1}^{n} s_i = \sum_{i=1}^{n} \sum_{j=1}^{t} S_i(\alpha_j)\lambda_j = \sum_{j=1}^{t} \lambda_j \sum_{i=1}^{n} S_i(\alpha_j) = \sum_{j=1}^{t} s'_j$$

Where S_i denotes Shamir polynomials of each participant, and α_i their Shamir public points. s' are the t-out-of-t reconstructed additive keys by the t participants.

F FairFed's algorithmic description

This section provides an overview of the the aggregation algorithm of `FairFed`.

The algorithm starts by collecting general statistics about the union dataset using individual statistics on clients' datasets. These measurements enable the computation of a global fairness measure F_g^t at every round. Subsequently, clients locally update the global model (LocalUpdate) and transmit the required metrics to the server. Along with the fairness-aware aggregation on the server side, clients continuously perform a *local debiasing* [38, 80] before computing their local update. The m_k^t components are individual statistics on each client's update measured on its local dataset and collectively sum to the global fairness measure F_g^t. Their definition depends on the considered fairness metric.

Algorithm 1 `FairFed` [30]

Require: A pool of data-holders (clients)
Ensure: A fair global model $\mathcal{M}_{\theta_g}$

Initialize global model parameters $\theta_g^{(0)}$
Aggregate union dataset statistics:
$\{P(Y=1, S=s_0), P(Y=1, S=s_1)\}$ from clients.
for $t = 1$ to T **do**
  for $i = 1$ to n **do** ▷ clients in parallel
    $\theta_i^t, F_i^t, m_i^t \leftarrow \text{ClientLocalUpdate}(\theta_g^t)$
  end for
  $F_g^t \leftarrow \sum_{i=1}^{n} m_k^t$
  $\bar{\omega}_i^t \leftarrow \exp(-\beta \cdot \mid F_i^t - F_g^t \mid) \cdot \frac{n_i}{\sum_{k=1}^{n} n_k} \quad \forall i \in [n]$
  $\omega_i^t \leftarrow \frac{\bar{\omega}_i^t}{\sum_{i=1}^{n} \bar{\omega}_i^t} \quad \forall i \in [n]$ ▷ Weights normalization
  $\theta_g^{t+1} \leftarrow \sum_{i=1}^{n} \omega_i^t \theta_i^t$
end for
return θ_g^T

G FHE-friendly polynomial approximations

To comprehensively assess the impact of polynomial approximations on the utility and fairness of the global model, we conduct a consistent plaintext FairFed workflow (Adult with identical dataset partitioning, local debiasing techniques and identical β values) using various polynomial approximations of the exp functions as described in [30], which we compare to our degree 2 approximation provided in 5. The considered polynomial approximations are inspired by FHE literature [18, 48]:

(1) Taylor expansion (degrees 2 and 4):

$$\exp(x) \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{8} + \frac{x^4}{24}$$

(2) Chebyshev approximation (degrees 2 and 4):

$$\exp(x) \approx 2.2772 + 1.2195x + 0.4991x^2 + 0.0948x^3 + 0.0296x^4$$

Table 14 presents the obtained results.

While the degree 2 Taylor and Chebyshev approximations relatively preserve the final model's utility, [18] they fail to maintain its fairness. This loss in the fairness benefit is due to the polynomial approximations' low accuracy near zero, which undermines the intended rationale for weight assignment in [30]. The degree 4 Taylor and Chebyshev approximations demonstrate improved fairness in the global model compared to their degree 2 counterparts. This improvement is comparable to our approximation, as the accuracy of these approximations near zero increases, more closely replicating the weight assignment mechanism described in [30].

Recall that the weight assignment mechanism in [30] involves computing $\exp(-\beta|F_k^t - F_g^t|)$. Therefore, in addition to their higher multiplicative depth compared to our approach (2 instead of 1), the full weight assignment mechanism using these degree 4 approximations is challenging to compute homomorphically because the exp

[17] $sk = \sum_{i\in[n]} sk_i$, refers to encryption schemes with collaborative decryption a mechanism. The decryption consists of a product of the secret with the ciphertext is compatible with addition $< s, \sum_{i\in[n]} sk_i >= \sum_{i\in[n]} < sk_i, c >$

[18] This is due to weight normalization, which tends to smooth out abnormal values from the polynomial approximations and helps stabilize the training process.

Table 13: Average-case noise growth from [23]

Operation	Noise variance
Encoding	$\sigma^2_{\text{encode}} = \sigma^2 + \frac{1}{12}$
Encryption	$\sigma^2_{\text{fresh}} = (\|\|v\|\|_2^2 + \|\|s\|\|_2^2 + 1)\sigma^2_{\text{init}}$
Addition	$\sigma^2_{\text{add}} = \sigma_1^2 + \sigma_2^2$
Const. addition	$\sigma^2_{\text{plain\_add}} = \sigma^2$ (No growth)
Multiplication (without Relin. & Rescaling)	$\sigma^2_{mult} = N\sigma_1^2\sigma_2^2 + \sigma_1^2\|\|m_2\|\|_1^2 + \sigma_1^1\|\|m_1\|\|_1^2$
Square	$\sigma^2_{\text{square}} = 2N\sigma^4 + 4\sigma^2\|\|m\|\|_2^2$
Const. (c) Multiplication	$\sigma^2_{\text{plain\_mult}} = c^2\sigma^2$
Relinearization	$\sigma^2_{\text{relin}} = (\frac{1}{12}P^{-1}Nq_l^2\sigma^2) + \mathbb{1}_{P\nmid q_l}\sigma^2_{\text{round}}$
Rounding	$\sigma^2_{\text{round}} = \frac{N}{18} + \frac{1}{12}$
Full multiplication	$\sigma_{\text{full\_mult}} = \frac{1}{\Delta^2}(\sigma^2_{\text{mult}} + \sigma^2_{\text{relin}}) + \sigma^2_{\text{round}}$

NB: σ^2 denotes input ciphertext's noise variance. For arithmetic operations involving two operands, σ_1^2 and σ_1^2 denote first and second operand's noise variance respectively, and σ^2_{init} refers to the initial RLWE Gaussian noise variance, and is part of the scheme's parameter-set.

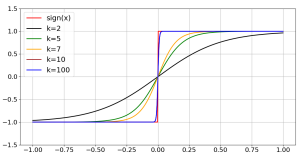
Table 14: Plaintext FairFed performance after 300 learning iterations with various approximations of exp**.**

Measures	Taylor-2	Cheb-2	Taylor-4	Cheb-4	Ours
Acc	0.8309	0.8403	0.8240	0.8515	0.8419
Precision	0.8812	0.8560	0.8215	0.8610	0.8327
Recall	0.6612	0.7631	0.7189	0.7134	0.7377
EOD	0.1474	0.1638	0.0492	0.0134	0.0104
SPD	-0.1793	-0.1427	-0.0970	-0.1203	-0.0911

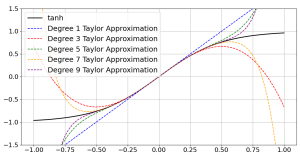
function in [30] is applied to the absolute value representing the distance between F_k^t and F_g^t. In contrast to our approach, these approximations of exp include odd-degree terms. Hence, the absolute value must be computed homomorphically, which requires using sign function approximations [10, 20, 41, 52, 59] ($|x| = \text{sign}(x) \cdot x$). To achieve this, prior works [10, 20] exploit the similarities between the sign function and tanh and use Taylor approximations of tanh. In contrast, our approach avoids this due to the absence of odd-degree terms, allowing the absolute value to be naturally incorporated.

To further assess the effects of approximating the absolute value on the models' utility and fairness, we again conduct consistent plaintext FairFed experiments (similar dataset partitioning, local debiasing and β value) using a degree 4 Taylor expansion of exp, along with polynomial approximations of the sign function of various degrees. Similarly to [36] we rely on the Taylor approximation of $\tanh(kx)$ to compute the absolute value using the sign function. Figure 9 illustrates these approximations. Additionally, Table 15 presents the obtained results, in terms of utility and fairness, of composing the previous experiments (Table 14) using a degree-4 Taylor approximation of exp with several other Taylor approximations of the sign function to compute the full weight assignment mechanism of [30].

Besides the cumbersome computational overhead, the use of two composed polynomial approximations (to compute both exp and tanh) strongly harms the accuracy of the weight assignment mechanism, which in turn leads in a clear degradation of the fairness benefits of FairFed as observed in Table 15. Regarding the approximation of the sign function for computing the absolute value, the



Proximity of $\tanh(kx)$ **to** $\text{sign}(x)$ **as** k **increases.**



Taylor approximations of $\tanh(kx)$ **with** $k = 2$**.**

Figure 9: Polynomial alternatives of the sign **function.**

Table 15: Plaintext FairFed performance after 300 learning iterations with various approximations of $\tanh(kx)$ **with** $k = 10$ **as** sign **function and degree 4 Taylor expansion of** exp**.**

Measures	$d = 1$	$d = 3$	$d = 5$	$d = 7$	$d = 9$
Acc	0.8431	0.8211	0.8136	0.8531	0.8401
Precision	0.7991	0.8330	0.7422	0.7903	0.7911
Recall	0.8602	0.7798	0.8987	0.8551	0.8593
EOD	0.1522	0.1472	0.1652	0.1531	0.1634
SPD	0.1899	0.1699	0.1789	0.1312	0.1586

polynomial approximations are accurate only within a small interval around 0, specifically when F_k^t is close to F_g^t. When F_k^t deviates significantly from F_g^t, the quality of the polynomial approximation deteriorates sharply, leading to anomalous values (greater than 1 or less than -1). These anomalous values are then used as input to the polynomial approximation of exp, exacerbating the anomalies and ultimately rendering the homomorphically cumbersome weight assignment mechanism ineffective. This is observed in fairness levels (EOD & SPD) of the final global model being equivalent (or even worse) to those obtained from a centralized, or FL training, without any fairness intervention.

H FHE overhead

In this section, we provide the computational cost of the key steps of our approach in both plaintext (without the threshold FHE encryption layer) and using the threshold FHE encryption. The FHE

overhead of each step denotes the difference between a plaintext execution and an execution on CKKS-encrypted data (with the specified parameters). More precisely, 100% FHE overhead indicates that the runtime without the FHE-encryption layer is either negligible or nonexistent while 0% overhead indicates identical costs.

Table 16: Step-wise runtimes (in seconds), and FHE computational overhead with $n = 100$, $t = 25$ and $\log(N) = 12$.

Steps	Plaintext	With FHE	FHE Overhead (%)
Threshold Setup	-	19	100%
LocalUpdate	53	53	0%
Encryption	-	$3 \cdot 10^{-3}$	100%
Weights computation	$7 \cdot 10^{-6}$	1.2	$\approx$ 100%
Aggregation	$3 \cdot 10^{-3}$	1.8	$\approx$ 100%
Collab. decryption	-	1.7	100%
Total ($T = 300$)	15900	16800	5.3%

The threshold keys setup time being a non-iterative step, it is amortized on the total number of training rounds (300). Encryption time represents the CKKS encoding and the canonical RLWE encryption and results in a very small computational overhead. The collaborative decryption consists of a nearly negligible local computation[19] and the runtime of this step is primarily induced from the network latency. The local update step dominates the runtime of each iteration. For large models and complex datasets, this step accounts for a greater share of the computational load, making the FHE overhead appear even less significant compared to the local update.

I Communication overheads

Table 17 reports the bandwidth usage and the communication overhead with respect to 3 network settings defined as follows:

- Setting 1 — Low latency (5ms), high bandwidth (1$Gbps$).
- Setting 2 — High latency (100ms), low bandwidth (100$Mbps$).
- Setting 3 — Medium latency (50ms), medium bandwidth (500$Mbps$).

Note that the network performances' results introduced in Table 7 refers to Setting 3.

[19]Each client subtracts its partial secret key share from the ciphertext as in Appendix E.2.

Table 17: Data size, bandwidth consumption (in GB), and communication overhead (in seconds) for the collaborative decryption phase of a single ciphertext encrypting $\frac{N}{2}$ model parameters, with $\log_2(Q) = 60$.

$\log_2(N)$	Active parties t	Setting 1		Setting 2		Setting 3	
		Bandwidth (GB)	**Comm. (s)**	**Bandwidth (GB)**	**Comm. (s)**	**Bandwidth (GB)**	**Comm. (s)**
11	3	0.09	0.01	0.09	0.05	0.09	0.03
	5	0.30	0.02	0.30	0.10	0.30	0.06
	10	1.38	0.05	1.38	0.22	1.38	0.13
	15	3.22	0.08	3.22	0.48	3.22	0.30
	20	5.83	0.10	5.83	0.72	5.83	0.45
12	3	0.18	0.02	0.18	0.08	0.18	0.05
	5	0.61	0.04	0.61	0.16	0.61	0.10
	10	2.76	0.10	2.76	0.44	2.76	0.28
	15	6.45	0.16	6.45	0.96	6.45	0.60
	20	11.67	0.22	11.67	1.40	11.67	0.90
13	3	0.37	0.04	0.37	0.14	0.37	0.09
	5	1.23	0.08	1.23	0.22	1.23	0.15
	10	5.53	0.18	5.53	0.50	5.53	0.34
	15	12.90	0.28	12.90	1.10	12.90	0.75
	20	23.35	0.40	23.35	1.60	23.35	1.00
14	3	0.74	0.08	0.74	0.18	0.74	0.12
	5	2.46	0.16	2.46	0.30	2.46	0.20
	10	11.06	0.36	11.06	0.66	11.06	0.45
	15	25.80	0.54	25.80	1.10	25.80	0.75
	20	46.69	0.80	46.69	1.60	46.69	1.05
15	3	1.47	0.10	1.47	0.25	1.47	0.16
	5	4.91	0.20	4.91	0.40	4.91	0.30
	10	22.11	0.45	22.11	0.85	22.11	0.60
	15	51.60	0.65	51.60	1.25	51.60	0.90
	20	93.34	0.90	93.34	1.60	93.34	1.20