future internet

Review

Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm

Rezak Aziz [1,*], Soumya Banerjee [1,*], Samia Bouzefrane [1] and Thinh Le Vinh [2]

[1] CEDRIC Lab, Cnam, 292 rue Saint Martin, 75003 Paris, France; samia.bouzefrane@lecnam.net
[2] Faculty of Information Technology, Ho Chi Minh City University of Technology and Education, Thu Đuc, Ho Chi Minh City, Vietnam; thinhlv@hcmute.edu.vn
* Correspondence: rezak.aziz@lecnam.net (R.A.); soumya.banerjee@lecnam.net (S.B.)

Abstract: The trend of the next generation of the internet has already been scrutinized by top analytics enterprises. According to Gartner investigations, it is predicted that, by 2024, 75% of the global population will have their personal data covered under privacy regulations. This alarming statistic necessitates the orchestration of several security components to address the enormous challenges posed by federated and distributed learning environments. Federated learning (FL) is a promising technique that allows multiple parties to collaboratively train a model without sharing their data. However, even though FL is seen as a privacy-preserving distributed machine learning method, recent works have demonstrated that FL is vulnerable to some privacy attacks. Homomorphic encryption (HE) and differential privacy (DP) are two promising techniques that can be used to address these privacy concerns. HE allows secure computations on encrypted data, while DP provides strong privacy guarantees by adding noise to the data. This paper first presents consistent attacks on privacy in federated learning and then provides an overview of HE and DP techniques for secure federated learning in next-generation internet applications. It discusses the strengths and weaknesses of these techniques in different settings as described in the literature, with a particular focus on the trade-off between privacy and convergence, as well as the computation overheads involved. The objective of this paper is to analyze the challenges associated with each technique and identify potential opportunities and solutions for designing a more robust, privacy-preserving federated learning framework.

Keywords: federated learning; differential privacy; homomorphic encryption; privacy; accuracy

check for updates

Citation: Aziz, R.; Banerjee, S.; Bouzefrane, S.; Le Vinh, T. Exploring Homomorphic Encryption and Differential Privacy Techniques towards Secure Federated Learning Paradigm. *Future Internet* **2023**, *15*, 310. https://doi.org/10.3390/fi15090310

Academic Editors: Qiang Duan, Zhihui Lu and Claude Chaudet

Received: 3 June 2023
Revised: 23 August 2023
Accepted: 7 September 2023
Published: 13 September 2023

1. Introduction

1.1. Background

The trends of advanced internet applications have had an overwhelming impact, particularly with the introduction of numerous machine learning (ML) algorithms. These algorithms have exhibited immense potential for a wide range of real-world applications. However, the success of these applications relies heavily on the establishment of a trust and secure paradigm. According to Gartner investigations [1], it is predicted that, by 2024, 75% of the global population will have their personal data covered under privacy regulations. Without this foundation trust and security, the future internet and digital economy, with their unlimited potential, will always be underestimated. To address the security concerns in federated environments—including the inherent dilution of the internet among mass usage, common vulnerabilities stemming from the Internet of Things, identity authentication, and significant digital fragmentation—various isolated and separate algorithms have been developed. However, these algorithms have proven insufficient.

The performance of machine learning algorithms depends on access to large amounts of data for training. In traditional machine learning, training is centrally held by one organization that has access to the whole training dataset. In practice, data are often distributed across multiple parties, and sharing it for training purposes is not simple due to privacy policies and regulations like the General Data Protection Regulation [2]. These regulations impose strict rules about how data can be shared and processed between organizations.

Due to these factors, federated learning (FL) has become a hot research topic on machine learning since its emergence on 2016 [3]. This promising technique allows multiple parties to jointly train a global model by only exchanging updates about local models and without the need to share their private datasets. This offers a promising solution to mitigate the potential privacy leakage of sensitive information about individuals. Recent works have demonstrated that FL may not always provide privacy and robustness guarantees [4–14]. While the private data never leaves their owner, the exchanged models are prone to memorization of the private training dataset. Some sensitive information may be inferred from the shared information using some well-known attacks like gradient inversion, reconstruction attacks, membership inference, and property inference attacks.

One way to mitigate this type of attacks is to use privacy preserving techniques like differential privacy (DP) and homomorphic encryption (HE). Differential privacy offers a way to disrupt data while preserving the statistical properties of the data. This allows us to have meaningful analysis and statistics while countering some of previous attacks. On the other hand, homomorphic encryption allows for conducting computation on encrypted data and then decrypting only the result. This allows FL to access the aggregation of gradient without accessing the gradient themselves.

Each technique has its own advantages and limitations. In this paper, we focus on the different works of the literature that use HE and DP techniques in federated learning context. We aim to analyze the advantages and the limitations of each technique taken alone before addressing the combination of the two techniques.

1.2. Motivation

A plethora of research efforts have examined privacy concerns in federated learning. These studies encompassed various aspects and topics including foundational concepts [15–17], identification of threats and corresponding solutions [18–21], exploration of privacy techniques [22,23], and applications within healthcare [24,25], as well as communications and mobile networks [26,27]. The highlights and the key concepts included in these studies are listed in Table 1. While these works offer comprehensive surveys of techniques, they often do not delve into the detailed application of these techniques as evidenced in the literature. Furthermore, with the exception of [22], these studies have largely overlooked the hybrid application of privacy methods where multiple techniques are employed in concert.

Table 1. Comparison with related surveys.

Ref	Year	Attacks	Defenses			Detailed Methods and Strategies
			DP	HE	Hybrid (DP + HE)	
[17]	2019		✓	✓		
[18]	2020	✓				
[23]	2020		✓	✓		
[24]	2020	✓	✓	✓		
[26]	2020	✓	✓	✓		
[21]	2020	✓	✓	✓		
[20]	2021	✓	✓	✓		✓
[16]	2021	✓	✓	✓		
[22]	2022	✓	✓	✓	✓	
[19]	2023	✓	✓			
[25]	2023		✓	✓	✓	
Ours	2023	✓	✓	✓	✓	✓

The authors of [22] did acknowledge different combinations of techniques in their work, but they did not closely examine the specific methods by which these techniques are employed. In contrast, our paper focuses on the intersection of homomorphic encryption (HE) and differential privacy (DP) within the framework of federated learning. The factors and parameters we take into account while comparing our work with previous studies are the specific attacks and defenses they discuss. Another aspect we consider is how thoroughly they explain their methods and strategies. We delve into the advantages and drawbacks of combining these techniques, as we think that this combination could lead to better privacy-preserving federated learning systems. This would allow us to leverage the unique strengths inherent to each individual technique.

1.3. Contribution

Our paper makes a notable contribution by thoroughly exploring and examining various scholarly sources. The primary scope of this paper revolves around addressing privacy concerns in federated learning. Consequently, certain related issues, such as communications, systems heterogeneity, and statistical heterogeneity, are intentionally excluded from our focus. Within the realm of privacy preservation, our main emphasis lies on exploring and analyzing differential privacy (DP) and homomorphic encryption (HE) techniques. While we acknowledge the existence of other privacy techniques in the literature, such as anonymization, secure multi-party computation, and blockchain, we do not directly delve into them in this paper. By narrowing our focus to DP and HE techniques, we can provide a more detailed and comprehensive analysis of their capabilities and limitations in the context of privacy preservation. This approach allows us to deliver a focused and valuable contribution to the research community and promotes a deeper understanding of the pivotal role these techniques play in ensuring secure and privacy-aware federated learning systems.

The main contributions of this paper are summarized as follows:

1. We scrutinize the array of research addressing privacy-related attacks in federated learning (FL), demonstrating the practicality and real-world relevance of these threats, highlighting their potential implications in distributed learning environments. Our primary focus lies on privacy attacks, where we delve into various techniques that adversaries can employ to compromise the privacy and security of FL systems.
2. We delve into the role of differential privacy (DP) in FL, detailing its deployment across various settings: central differential privacy (CDP), local differential privacy (LDP), and the shuffle model. By providing a comprehensive analysis of these DP deployment settings, we offer insights into the strengths, limitations, and practical implications of each approach.
3. We investigate the application of homomorphic encryption (HE) as a powerful tool to enhance privacy within FL. Our primary focus is on countering privacy attacks and safeguarding sensitive data during the collaborative learning process. Through our investigation, we provide valuable insights into the capabilities and limitations of homomorphic encryption in FL.
4. We examine the body of research that explores the fusion of homomorphic encryption (HE) and differential privacy (DP) in the context of federated learning (FL). Our primary objective is to shed light on the motivations behind such integrations and understand the potential benefits they offer in enhancing privacy and security in distributed learning environments.

The rest of this paper unfolds as follows: Section 2 provides essential background knowledge on HE, DP, and FL. Section 3 delves into various privacy attacks within the FL framework. Section 4 discusses the combination of DP with FL, while Section 5 explores the use of HE for protecting privacy. In Section 6, this paper explores the combined use of HE and DP, emphasizing the potential benefits of this fusion. Section 7 is dedicated to the discussion of the results, offering deeper insight into our findings. Finally, Section 8 presents our conclusions and proposes directions for future research.

2. Preliminaries

The key concepts that we treat in our paper are federated learning, differential privacy, and homomorphic encryption. Here, we give an overview of the different techniques.

2.1. Federated Learning

The term federated learning (FL) was introduced in 2016 by McMahan et al. [3]. FL is a machine learning setting where many clients collaborate to train a centralized ML model. Each client's raw data are stored locally and not exchanged with other parties; only updates needed for immediate aggregation are shared with the central server.

Two main settings are discussed in the literature [16]: the cross-device and the cross-silo. The difference between the two is simple, cross-device is associated with mobiles and IoT devices while cross-silo is associated with organizations like hospitals, banks, etc. In cross-silo, the number of clients is small and they have large computational ability. On the other hand, cross-device considers a huge number of clients with small computation power. Another difference between the two settings is reliability. In cross-silo, the organizations are always available to train, unlike user devices.

FL can also be classified by data partition. We distinguish Horizontal FL, Vertical FL and Hybrid FL. In HFL, the datasets of the clients have the same features space. In VFL, the local datasets have the same individuals, but with different features. The hybrid setting is a combination between HFL and VFL.

A typical federated training process is considered by the algorithm of FedAvg proposed by McMahan et al. [3]. It consists of five steps: The server selects a subset of clients according to some criteria. The selected client downloads the current model weights and a training program from the server. Each client locally computes an update to the model by executing the training program. The server then collects an aggregate of device updates and updates the central model.

2.2. Differential Privacy

Differential privacy is a widely used standard to guarantee privacy in data analysis. The main idea of DP is to consider a thought experiment in which we compare how an algorithm behaves on a dataset D_1 with the way it behaves on a hypothetical dataset D_2, in which one person's record has been removed or added. These two datasets are considered "neighbors" in the dataset space. Hence, we say that an algorithm is differentially private if running the algorithm on two neighboring datasets yields roughly the same distribution of outcomes. In other words, differential privacy ensures that the outcomes of M are approximately the same whether or not the person i joins the dataset. Formally, DP is defined by Dwork et al. in 2006 [28] as follows.

Definition 1. *A randomized function M gives (ϵ, δ)-differential privacy if for all datasets D_1 and D_2 differing on at most one element, and all $S \subset Range(M)$,*

$$Pr[M(D_1) \in S] \leq e^{\epsilon} \times Pr[M(D_2) \in S] + \delta \tag{1}$$

In the Definition 1, ϵ is a non-negative real number that determines the level of privacy protection provided by the algorithm. A lower value of ϵ corresponds to a stronger privacy guarantee. The value of δ is a small positive real number that represents the probability of any failure of the definition. When δ is set to 0, it is referred to as pure differential privacy.

Three major properties arise directly from this definition: composition, post-processing, and group privacy. These properties are the key to design powerful algorithm from basic mechanisms:

- Composition: offers a way to bound privacy cost of answering multiple queries on the same data.
- Post-processing: ensures that the privacy guarantees of a differential privacy mechanism remain unchanged even if the output is further processed or analyzed.

- Group privacy: this definition can be extended to group privacy by considering two datasets differing on at most k records instead of 1 record.

As stated in the definition, DP is a property of an algorithm M. There are several methods to achieve DP based on adding noise to the input data, the output data, or the intermediate result. The noise can be generated using different mechanisms, the well-known ones are the Laplace mechanism, the Gaussian mechanism, and the exponential mechanism.

Two main settings are discussed in the literature for differential privacy: the centralized DP (CDP) and the local DP (LDP). In CDP, the noise is added by a centralized server that collects first the data then applies the mechanism. In LDP, the noise is added at the client level before collecting the data. LDP offers stronger privacy guarantees, as the noise is added closer to the source of the data. Additionally, a hybrid setting called the shuffle model is also explored in the literature. The shuffle model aims to combine the benefits of both CDP and LDP. In this setting, privacy is enhanced through anonymization achieved by shuffling the data. The noise is added centrally by a shuffler before passing the data to the analyst server, which enables the system to attain the performance advantages of CDP while maintaining the privacy guarantees provided by LDP.

2.3. Homomorphic Encryption

Homomorphic encryption is a cryptographic primitive that allows third parties to perform arithmetic operations on ciphertexts without decrypting them. It provides the same result as encrypting after operating in cleartext messages.

More formally, an encryption scheme is called homomorphic over an operation $*$ if it supports the following property:

$$E(m_1) * E(m_2) = E(m_1 * m_2)$$

where E is the encryption algorithm and m_1, m_2 belong to M the set of all possible messages.

An HE scheme consists of four algorithms [29]: *KeyGen*, *Enc*, *Dec*, and *Eval*. *KeyGen* generates a pair (public key, private key) for the asymmetric configuration and a secret key for the symmetric version. *Enc* is the encryption algorithm and *Dec* is the decryption algorithm.

While the three algorithms (*KeyGen*, *Enc*, *Dec*) are common to conventional cryptosystems, an additional algorithm is needed for homomorphic encryption schemes, called the *Eval* algorithm. This algorithm is defined as follows:

$$Eval(f, C_1, C_2) = f(m_1, m_2)$$

where $Dec(C_1) = m_1$, $Dec(C_2) = m_2$ and f is a function that can be addition or multiplication.

Based on the number (limited or unlimited) and the type of operation (addition or multiplication), HE is classified into three types of schemes: Partially Homomorphic encryption (PHE), Somewhat Homomorphic Encryption (SWHE), and Fully Homomorphic Encryption (FHE). PHE allows only to perform one type of operation on unlimited way. When the operation is addition, like in the Paillier Scheme [30], we say it is an Additive Homomorphic Encryption (AHE). When it is the multiplication, we say that it is a multiplicative scheme, like in the RSA scheme.

SWHE allows for both operations, but the number of operations is limited. On the other hand, FHE allows making unlimited operations of both types. This type of scheme was possible after the Gentry breakthrough in 2009 [31].

3. Privacy Attacks in FL

While federated learning (FL) is generally regarded as a privacy-preserving technique in machine learning, recent studies have revealed a potential privacy concern (see Figure 1). This concern arises from the fact that, although FL avoids the need to share private client datasets during the learning process, the exchange of gradients in FL can inadvertently disclose sensitive information about the client's private data. This issue is particularly pronounced in FL due to the large number of participants involved and the inherent white-box setting of the FL framework. An insider may exploit the exchanged gradients to perform powerful attacks using passive strategy (one that doesn't influence the learning process) or an active approach (where they actively influence the learning process)such as conducting membership inference or launching a reconstruction attack. In this section, we will see the different attacks on privacy in FL. The goal and the vulnerabilities exploited by the adversary in these attacks are presented in Table 2.

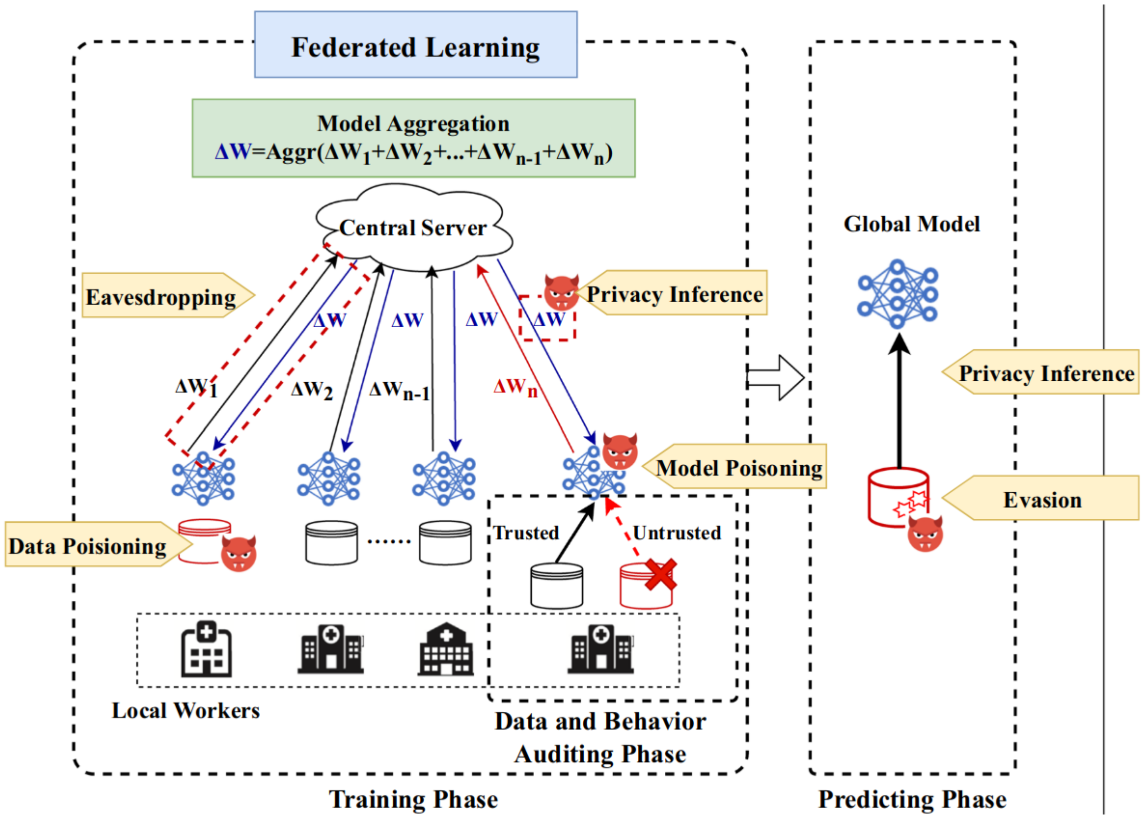

Figure 1. Attacks on the federated learning process [32].

3.1. Membership Inference

Membership inference is a type of attack in machine learning that aims to figure out whether a target data point is used to train a certain ML model. More formally, given x as the target point, M as a trained model, and some external knowledge K, this attack can be defined by the following function

$$A : x, M, K \rightarrow \{False, True\}$$

Here, this function returns *True* if the target x is in the training dataset and *False* otherwise. This attack can be made either in a black-box setting, where the attacker has only access to an API of the model M, or in a white-box setting, where the attacker has access to the whole model.

As we can remark, the attack model A is a binary classifier, and it can be constructed using different ways.

The first membership inference attack against machine learning models was presented by Shokri et al. in 2017 [33]. In this work, they consider a black-box setting and try to exploit the fact that ML models have a different behavior on the data they were trained on and the data that they see for the first time. In other words, they are modeling the membership inference problem as a problem of binary classification and try to train a model that distinguishes members from non-members of the target model. The idea of this work is to train the attack model using a shadow training technique (that was introduced in this work). They construct multiple shadow models that mimic the behavior of the target model, but for which the training set is known and have the same structure. Shokri et al. proposed some methods to generate the shadow training data and then the method used to train the attack model. The proposed attack was evaluated against neural networks using different datasets: CIFAR, Purchases, Locations, Texas hospital stays, MNIST, and UCI Adult. The authors confirm that their results show that models created using MLaaS can leak a lot of information about their training datasets. The assumptions considered by Shokri et al. are considered strong, which reduce the scope of the membership inference attack [34]. This motivated Salem et al. in 2019 [34] to propose relaxations of these assumptions. They showed that relaxing the number of shadow models to one shadow model and assuming that the shadow training data are distributed similarly to target training data produce performances that are similar to those provided by Shokri et al. [33]. Furthermore, considering an attack model that is independent of the training data distribution may also reveal some information, but this is not as efficient as in the previous work.

The two previous works focus on general machine learning (ML) models, but it is important to consider the unique characteristics of federated learning, which may present a larger attack surface [35]. Pustozeova and Mayer in 2020 [35] demonstrated that membership attacks performed by an insider in a sequential federated learning setting are more effective compared to centralized settings. Unlike the work of Shokri et al. [33] that examines black-box attacks, in the federated learning setting, an insider attacker has knowledge of the model's architecture, making the attack more efficient. Additionally, if multiple insider attackers collaborate, the attack can become more sophisticated. Hu et al. [36] further assert that, through Membership Inference Attacks (MIA), adversaries can even infer which participant possesses the data. The authors demonstrate that an honest-but-curious server can estimate the data source without violating federated learning protocols.

In 2019, Nasr et al. [37] introduced a comprehensive Membership Inference Attack (MIA) that targets the privacy vulnerabilities of the stochastic gradient descent (SGD) algorithm within the context of federated learning. Their study focused on the white-box setting, wherein the attacker has access to the model's loss and can compute the gradients of the loss with respect to all parameters using a simple backpropagation algorithm. The authors demonstrated that, in deep neural networks, the distribution of the model's gradients on members can be distinguished from the distribution of non-members. They explained that the initial layers of the neural network tend to contain less individual-specific information, requiring the attacker to devise specific attacks for each layer. The attack model proposed by Nasr et al. consisted of feature extraction components and an encoder component. To extract features from the output of each layer, they employed a fully connected network, incorporating the one-hot encoding of the true label and the loss. For the gradients, a convolutional neural network was utilized. The output from this step was then fed into an FCN encoder, which provided the membership probability of the input. Through experimentation with various datasets such as CIFAR100, Texas100, and Purchase100, the authors demonstrated that even well-generalized models are highly susceptible to white-box membership inference attacks.

Gu et al. in 2022 [38] proposed a membership inference attack named CS-MIA, which utilizes prediction confidence series (PCS) in federated learning. This attack takes advantage of the observation that the prediction confidence on training and testing data exhibit distinct changes over rounds in federated learning. The authors demonstrated that the variations of models across rounds in federated learning can be leveraged to differentiate between members and non-members of the target model. They trained a fully connected network to process the PCS and learn the discrepancies between training and testing data. The researchers designed membership inference methods for both local and global attackers and introduced an active global attack to enhance attack performance. To train the attack model, the authors drew inspiration from the shadow training technique introduced by Shokri et al. [33]. They generated shadow confidence series for member and non-member instances by involving members in a federated learning process. Subsequently, they computed the confidence of the shadow model on both member and non-member instances, using this information to train the attack model. Experimental results highlighted the vulnerability of federated learning privacy to the differences between training and testing confidence series. CS-MIA achieved a membership inference accuracy of over 90% on various benchmark datasets, indicating a significant threat to the privacy of federated learning.

3.2. Class Representatives Inference

The inference of class representatives tries to extract generic class representatives of the global data rather than the real data in the training datasets [39]. This is similar to the concept of model inversion attack, proposed by Fredrikson et al. in 2015 [40]. In the special case where all class members are similar, the result of this attack is similar to the training data. For example, in a facial recognition model, each class corresponds to a single individual, and the output of this attack is similar to any image that represent this person.

In the realm of federated learning, the utilization of Generative Adversarial Networks (GANs) allows the execution of such attacks. Hitaj et al. in 2017 [41] designed such an attack in federated learning using GAN.The attacker, acting as an honest-but-curious client in the federated learning topology, tries to influence the learning process. In their work [41], the attacker locally trains a GAN model capable of generating synthetic samples that look like the samples from the victim's data but that are supposed to belong into a different class. In this way, the victim will work harder to distinguish the fake class from his class and reveal more information about his dataset. The experiment results showed that this attack is effective and generate representative samples of training datasets.

Furthermore, in 2018, Wang et al. [39] expanded the scope of the attack to breach client-level privacy. While acknowledging the effectiveness of the GAN-based attack proposed by Hitaj et al. [41], the authors observed that the adversarial influence of the client could alter the architecture of the shared model. Moreover, they considered a powerful malicious client in their analysis. In order to overcome these limitations, Wang et al. introduced a more practical and inconspicuous attack on the federated learning model, known as mGAN-AI. In contrast to the previous attack, which was conducted by the client, the authors of mGAN-AI assumed the presence of a malicious server. They went a step further in breaching client-level privacy by utilizing a GAN with a multitask discriminator. This discriminator not only performed the task of a standard GAN, but also distinguished the real data distribution of the victim from that of other clients. Through experimental evaluations on datasets such as MNIST and AT&T, the researchers demonstrated that mGAN-AI could reconstruct samples close to the victim's training samples.

3.3. Properties Inference

The properties inference attack was introduced by Ateniese et al. in 2013 [42]. It aims to extract some private statistical information about the training set. This statistical information is unexpected to be shared and might be irrelevant to the main training task. This type of attack violates the intellectual property of the model producer, while it can be used to perform more complex attacks that infer something private about the individuals. According to the findings presented in [42], the adversary can construct a meta-classifier capable of categorizing the target classifier based on the presence or absence of a specific property, denoted as P, that the adversary seeks to infer. In the context of the study, the application considered was the inference of the ethnicity of a population, specifically distinguishing between Indian and non-Indian individuals, utilized in the training process. To accomplish this, shadow classifiers were trained on the same task and using similar datasets as the target classifier, but constructed to possess or lack the property P. The parameters of these shadow classifiers were then employed to train the meta-classifier.

The work presented in [42] focuses on a centralized machine learning (ML) context using Support Vector Machines (SVM) and Hidden Markov Models (HMM). In contrast, Melis et al. in 2018 [43] were the first to explore the unintended feature leakage within collaborative machine learning and federated learning (FL). They demonstrated that the exchanged updates during the FL process can inadvertently disclose information about the participants' data. By exploiting this leakage, both passive and active property inference attacks can be executed to infer properties that are unrelated to the original task of the model. In this scenario, the adversary is a participant in the process of FL that exploits the periodic updates of the global model to perform the attack. The information leakage that can be exploited are the leakage from the sparse embedding layer, particularly for the non-numerical data such as in Natural Language Processing, and leakage from the gradients. The results suggest that leakage of unintended features exposes FL to powerful inference attacks.

Ganju et al. in 2018 [44] concentrated on inferring global properties of the training data by conducting a white-box attack against Fully Connected Neural Networks (FCNNs). Their goal was to deduce properties such as the data production environment or the proportion of data belonging to a specific class. Unlike the approach proposed by Ateniese et al. [42], which is not practical for FCNNs, Ganju et al. addressed the challenges posed by FCNNs, particularly the fact that permutations of nodes in each hidden layer can lead to equivalent FCNNs. This property makes it difficult for meta-classifiers. To overcome this challenge, Ganju et al. proposed two strategies: neuron sorting and set-based representation. These strategies enhance the effectiveness of the attack by ensuring better classification performance. The authors compared their results with the work of Ateniese et al. [42] and demonstrated the improved performance of their approaches when applied to FCNNs. The results underscore the challenge posed by FCNNs and highlight the effectiveness of the neuron sorting and set-based representation strategies in addressing this challenge.

Several works have studied models other than Fully Connected Neural Networks (FCNNs) to explore their vulnerability to property inference attacks. Zhou et al. [45] investigated generative models, particularly generative adversarial networks (GANs). Their work proposed a general attack pipeline applicable to both the full black-box and partial black-box settings. This research demonstrated the feasibility of conducting property inference attacks not only on discriminative models but also on generative models, highlighting the effectiveness of such attacks across both model types.

3.4. Training Samples and Labels Inference

Training samples and labels inference, also known as reconstruction attacks, aim to reconstruct the original dataset belonging to a client involved in the federated learning process. These attacks focus on recovering the training samples and their corresponding labels from the aggregated model. By exploiting the information present in the model's parameters or gradients, adversaries attempt to recreate the client's original dataset, potentially compromising the privacy and confidentiality of the client's data.

Zhu et al. [4] in 2019 demonstrated that it is possible to obtain the private training data from the publicly shared gradients. Their method, known as Deep Leakage from Gradient (DLG), utilizes an optimization algorithm to recover pixel-wise accurate information for images and token-wise matching for texts. The attack is performed by generating "dummy" inputs, then by performing the forward–backward pass, they compute dummy gradients from the global model. Instead of updating weights of the dummy model, they update the dummy inputs and labels by minimizing the distance between dummy gradients and real gradients. The results show that they can achieve exact pixel-wise data recovering using just the shared global model and local gradients.

Zhao et al. [5] observed that DLG is unable to extract the ground-truth labels. To address this limitation, they proposed a method called iDLG. They demonstrated that the signs of gradients of the classification loss with respect to correct and wrong labels are opposite. This enables to always extract the ground truth labels.

Geiping et al. [6] in 2020 state that previous works are based on Euclidean cost function with an optimization via L-BFGS. These choices may not be optimal for realistic architectures. The authors propose to use a cost function based on cosine similarity to catch more information about the data. They find that, if we decompose the gradient into its norm magnitude and its direction, then the magnitude captures only information about the state of the training while the direction can carry significant information about the change in prediction when taking a gradient step towards another data point. This approach aims to find images that pursue similar prediction changes, and it was the first work that pushed the boundary towards ImageNet-level gradient inversion.

Yin et al. [46] introduced in 2021 GradInversion to recover the individual images that a client possesses within a batch by optimizing the input data to match the gradients provided by the client. The main challenge is to identify the ground-truth label for each data point in the batch. The main contribution is the introduction of the group consistency regularization term by computing a registered mean image from all candidate images. This allows for the reduction of the variance of the candidates, hence improving the convergence towards the ground truth images.

Jin et al. in 2021 [9] affirmed that existing approaches do not scale well with large-batch data recovery and do not provide a strong theoretical justification on the capability of data recovery. Therefore, they designed CAFE (catastrophic data leakage in vertical federated learning), an advanced data leakage attack with theoretical analysis on the data recovery performance. The proposed algorithm consists of three steps: Recover the gradients of loss with respect to the outputs of the first FC layer, use the recovered gradients as a learned regularizer to improve the performance of the data leakage attack, and then use the updated model parameters to perform the data leakage attack. The experimental results demonstrate that CAFE can recover private data from the shared aggregated gradients while overcoming the batch limitation problem in previous attacks.

Ren et al. in 2022 [7] proposed a generative regression neural network (GRNN) to recover images from the shared gradient in FL. The attack recovers a private training image up to a resolution of 256*256 and a batch size of 256, which surpasses the previous state of the art. The proposed method addresses three major challenges in existing methods: model stability, the feasibility of recovering data from large batch size, and fidelity with high resolution. GRNN consists of a GAN model for generating fake training data and an FCN for generating the corresponding label. A fake gradient is generated given the shared model, and the two generators are optimized by approximating this fake gradient to the true gradient. The extensive experiments conducted by the authors show that their work outperforms DLG in terms of the addressed challenges.

Table 2. Privacy inference attacks against FL.

Ref	Year	Assumption Adversary	Active/Passive	Goal	Exploit
[41]	2017	Client	Active	Class representative inference	Influencing the learning process.
[39]	2018	Server	Active	Class representative inference	Influencing the learning process.
[37]	2019	Client	Active/Passive	Membership inference	Vulnerabilities of the SGD algorithm.
[38]	2022	Client/Server	Passive	Membership inference	Prediction confidence series.
[43]	2018	Client	Passive	Properties inference	Global model updates.
[44]	2018	Client	Passive	Global Properties inference	Shared gradients.
[4]	2019	Server	Passive	Training data inference	Shared gradients.
[5]	2020	Server	Passive	Training data inference	Shared gradients and their signs.
[6]	2020	Server	Passive	Training data inference	Shared Gradients and Cosine similarity.
[46]	2021	Server	Passive	batch data recovery	Gradient inversion.
[9]	2021	Server	Passive	Large batch data recovery	Shared aggregated gradients.
[7]	2022	Server	Passive	Training image recovery	Shared gradients.

4. Federated Learning with Differential Privacy

4.1. Role of DP in FL

As previously stated, DP is a powerful technique that gives strong mathematical guarantees for privacy protection. Differential privacy in the context of FL was explored at the early stages of FL by McMahan et al. in 2017 [47]. DP offers several benefits, including:

1. Protecting individual participant's data: DP achieves this by adding noise to the shared updates, thereby hiding the contributions of each individual in the FL process.
2. Protecting data against membership inference and reconstructions attacks: DP is known to be robust to this type of attacks.
3. Encouraging the user to participate in the learning process: DP provides strong privacy guarantees to the user by offering plausible deniability for them.
4. Facilitating compliance with regulations: DP offers a way for companies to comply with the requirements of various data protection regulations, such as the General Data Protection Regulation (GDPR).

In summary, DP in FL provides multiple advantages, including individual data protection, defense against privacy attacks, enhanced user privacy guarantees, and regulatory compliance support.

4.2. Related Works

The combination of DP with FL is an active research area. The main challenge in this field is the trade-off between privacy and utility. This challenge is addressed using the different setting in DP, including Centralized DP (CDP), Local DP (LDP), and the shuffle model. Table 3 presents the different selected works, presenting the key ideas and the shortcoming.

The use of CDP for federated learning was explored in 2017 by McMahan et al. [47]. They were the first to show that it is possible to train large recurrent language models with CDP. Their proposed algorithm is based on the FedAvg algorithm [3] and the moments' accountant technique [48], which provides tight composition guarantees for the repeated application of the Gaussian mechanism. The authors extended the FedAvg and FedSGD algorithms to provide differential privacy guarantees. Their findings show that achieving DP comes at the cost of increased computation rather than in decreased utility.

At the same time as McMahan et al. [47], Geyer et al. [49] investigated the use of CDP to protect participants' data from other malicious participants while considering the server honest-but-curious. They proposed an algorithm that aims to hide clients' contributions during the training while balancing the trade-off between privacy loss and model performance. The idea is to approximate the averaging of client models with a randomized mechanism. This mechanism involves random subsampling of clients, clipping the updates before transmission to the server, and distorting the clipped updates using a Gaussian mechanism before the aggregation. Experimental results show the feasibility of using CDP in FL; however, the number of clients has a major impact on the accuracy of the model.

In 2019, Choudhury et al. [50] studied the performance of CDP in healthcare applications using real-world electronic health data. They proposed to add noise to the objective function of the model instead of perturbing the data. They show that using differential privacy can lead to a significant loss in model performance for this kind of application.

Hu et al. in 2020 [51] emphasized that the research should focus on the trade-off between privacy loss and accuracy of the model. The authors proposed a privacy-preserving approach for learning personalized models on distributed data. Their approach consists of training a personalized model of each client using their local data but also the shared updates from other clients. They used a Gaussian mechanism to provide (ϵ, δ)-differential privacy guarantees for the shared gradient. The added noise is calibrated using the sensitivity of the updates. Hu et al. considered a threat model with an honest-but-curious server and malicious users. While evaluating their approach, they affirmed that it is robust to device heterogeneity and perturbation of noises, offering a good trade-off between accuracy and privacy.

The use of LDP has also been largely investigated in the context of FL. The motivation is that, contrary to CDP, LDP protects the user's data even from a malicious server and gives more flexibility to the clients to manage their privacy.

Bhowmick et al. in 2019 [52] investigated the use of LDP to defend against reconstruction attacks in FL. In this work, the privacy was provided through two steps. First, the LDP was employed at the client side to protect the private individuals' data. Then, in the server-side computation, the LDP was used to guarantee the privacy preservation of the global model update. This approach aimed to mitigate the reconstruction attacks while maintaining privacy and accuracy.

Liu et al. in 2020 [53] observed that applying LDP is challenging when the dimension of the data is large, as the injected noise is proportional to the number of dimensions. Additionally, a large batch size is needed to obtain acceptable accuracy. To overcome these challenges, the authors proposed a two-stage LDP privatization framework for federated stochastic gradient descent (SGD). In the first stage, they privately select the top k dimensions based on their contribution to the gradients. Their idea behind this stage is that not all the dimensions are equally important. While selecting the Top 1 dimension can be easily accomplished by the exponential mechanism, extending this to select k dimensions is more challenging. For this case, the authors proposed two alternative mechanisms. In the second stage, value perturbation using LDP is applied to ensure privacy while preserving the utility.

Ni et al. in 2021 [54] proposed an adaptive differential privacy federated learning model for medical IoT applications. Specifically, they proposed a DNN (named DPFLAGD-DNN) for adding noise to the model parameters according to the correlation between the model output and the characteristic of the training data. According to the authors, this method reduces the unnecessary noise and improves the accuracy. The process is that each client performs the model training according to the parameters obtained from the server and adaptively adds noise by DPFLAGD-DNN. After that, the noisy parameters are sent back to the server. Considering also the leakage from the down link, the authors proposed to add noise using the same mechanism in the server side before broadcasting the parameters. Experimental results show that the proposed algorithm can achieve high accuracy and may be more practical for medical IoT applications.

Sun et al. in 2021 [55] considers again the DNN in a DPFL setting. They addressed two main challenges, the fixed weight range assumptions in previous work and the privacy degradation due to high dimensionality of DNN. They proposed a new adaptive LDP mechanism according to the weight ranges of different DNN layers. They also proposed a shuffling mechanism for parameters to anonymize the data source. Here, the mechanism of shuffling considers the parameters and not the models. They assume that this is more efficient against side-channel linkage attacks than in the standard method of shuffling models.

The cross-silos setting of FL was considered by Chamikara et al. in 2022 [56]. The authors addressed the challenge of managing the noise and the privacy budget due to high dimensionality of parameter matrices in DNN. The method proposed by Chamikara et al. adds noise to the data input instead of the parameters. By considering a malicious clients and server, the noise is added in a specific manner. First, the clients locally train a conventional neural network (CNN) using their respective data and then use the convolutional module of the CNN to obtain flattened vectors of the input. These flattened vectors are then encoded into binary vectors. After that, the randomized response is applied as a DP mechanism to perturb the vectors before training the local deep neural network (DNN). Finally, the clients send their respective trained local models to the server for training the global model.

Shen et al. in 2023 [57] raised the issue in previous works that consider the same privacy's requirements for all clients. This approach fails to acknowledge that each client in the real world has unique privacy needs. The authors introduced a perturbation algorithm that enables personalized LDP. In other words, each client adjusts its privacy parameter ϵ_i according to the sensitivity of its data. The experimental analysis demonstrated that clients can adjust their privacy parameters while still maintaining high accuracy.

The shuffle model of DP was studied by Girgis et al. in 2021 [58–60]. Their research aimed to address the challenge of poor learning performance in LDP and tried to enhance the trade-off between privacy and utility. To achieve this, they propose to amplify the privacy by self-sampling and shuffling. The main contribution of their work lies on the concept of self-sampling. Contrary to the standard shuffle model where the server knows who the participants are, in this setting, the server does not have knowledge of the participant at each step. This approach avoids the need for coordination in participant selection during the federated process.

Table 3. Privacy-preserving FL using DP.

Ref	Year	DP Type	Key Idea	Trade-offs and Shortcomings
[47]	2017	CDP	Adding Gaussian noise by the server before global aggregation.	Increased computation cost and poor performance in non-IID setting
[49]	2017	CDP	Same as [47], but using subsampling of clients and clipping before sending updates.	The number of clients has a major impact on the accuracy of the model.
[50]	2019	CDP	Adding noise to the objective function instead of the updates.	Poor performance for healthcare applications
[51]	2020	CDP	Training a personalized model for each client using local data and the shared updates from other clients (Protected using DP).	Increased computation and communication cost
[52]	2019	LDP	Protecting local update from server using DP in the client side and protect global updates from clients using DP in the server side.	Increased computation cost
[61]	2020	LDP	Reducing noise injection by selecting the top k important dimension, then applying LDP.	Increased computation cost
[54]	2021	LDP	Adding adaptive noise to the model parameters using a deep neural network.	Increased computation cost
[55]	2021	LDP	Same as [54], but using adaptive range setting for weights and adding a shuffling step to amplify privacy	Increased computation cost
[56]	2022	LDP	Using the randomized response mechanism instead of the Gaussian and Laplacian mechanism.	Increased computation cost
[57]	2023	LDP	Using personalized privacy budget according to clients' requirements	The privacy budget is the same for all attributes.
[60]	2021	Shuffle	Amplifying privacy by self-sampling and shuffling. Real participants are unknown to the server.	Increased system complexity.

4.3. Discussion and Learned Lessons

The use of DP has been widely studied in federated learning using different settings, including CDP, LDP, and the shuffle model. One of the central challenges addressed in these settings is balancing between privacy and model performance.

CDP and LDP consider two primary adversaries: the clients and the aggregation server. CDP offers protection by safeguarding other clients' data from a malicious client while considering a trusted server. However, achieving this trust in practice can be challenging. Using LDP, on the other hand, eliminates the requirement to trust the server as the noise is added at the client level. However, this security comes at a cost to model performance.

The independent generation of noise by different clients in LDP adds substantial noise and requires more data to achieve the same level of accuracy as CDP.

Another issue by CDP and LDP is the anonymity of the clients. The server can track the source of updates, which widens the attack surface. The solution proposed to have the benefits of the two worlds of CDP and LDP while also guaranteeing anonymity is to use the shuffle model. In the shuffle model, the noise is generated by a shuffler, which also conducts the shuffling of updates to preserve the anonymity of the clients. The model can achieve a performance similar to CDP while not relying on a trusted server, as in LDP.

Many solutions have been designed in these different settings, going from designing new suitable mechanisms for DP to proposing alternative definitions of DP in the context of federated learning. It is also important to consider factors such as data distribution (vertical, horizontal, or hybrid) and the setting of FL (cross-device or cross-silo). Additionally, considering the correlation between the different attributes of the data is crucial. In fact, correlation is considered as a threat and may compromise the process of DP.

Furthermore, it is worth noting that DP alone may not counter all possible attacks. As a result, some works proposed to amplify DP by anonymization techniques. Other works also propose to amplify privacy by using other techniques such as secure multiparty computation and homomorphic encryption.

5. Federated Learning with Homomorphic Encryption

5.1. Role of HE in FL

Homomorphic encryption (HE) enables calculations over an encrypted domain, making it a good candidate for collaborative training of joint models in FL. HE can be applied in various ways within the FL framework, as seen in previous works.

One application of HE in FL is to hide client updates from the server. Instead of accessing the client's updates directly, the server will perform the aggregations in the encrypted domain and only access the final result. This approach provides an added security layer against eavesdropping and data breaches. By encrypting the updates, even if an unauthorized person intercepts the data, it will not have access to the raw data or the model updates.

Another way to utilize HE in FL is to collaboratively train the model without the need for intermediate decryption. In this scenario, the server conducts aggregations in the encrypted domain while having no method to decrypt the final result. Only clients having the decryption key can share the model.

HE can have other applications to counter adversarial attacks that do not deal with privacy, or auxiliary attacks that are facilitating privacy attacks, such as poisoning attacks. These attacks aim to compromise the integrity or reliability of the FL process. From security perspectives, to defend the server from model poisoning attacks, researchers have explored various variational measures. One such measure is CosDetect, proposed by Yaldiz et al. in 2023 [62], which employs a cosine similarity-based outlier detection algorithm to address fundamental issues more effectively than existing security solutions. The authors observed that the weight of the last layer pertaining to the local model update could be more sensitive to the local data distribution than other layers. This observation is significant, as it suggests that the last layer of local updates from malicious clients should exhibit outlier characteristics compared to updates from honest clients, making it more meaningful to a privacy attack. However, as this paper does not focus on such attacks, we will not delve deeper into them.

5.2. Related Works

The first-level combination of FL and HE has been initiated by researchers. The main purpose of HE in the context of privacy preserving is to safeguard the leakage of gradients, thereby by enabling secure aggregation during the learning process. Table 4 presents the different selected works, presenting the key ideas and the shortcoming.

Zhang et al. in 2020 [63] introduced BatchCrypt, a solution that reduces encryption and communication overhead when applying HE in cross-silo FL. The authors proposed a batch encryption technique where clients encode a batch of quantized values of gradients to a long integer and encrypt it. The main challenge addressed in their work is finding a feasible batch encryption scheme that allows direct summation of ciphertexts without intermediate decryption. To achieve this, they proposed a novel encoding technique using quantization of gradients. They adopt two complement representations with two sign bits, padding, and advanced scaling to avoid overflow. They also tackle the challenge of unbounded gradient by proposing an efficient analytical model (named dACIQ) for clipping. Compared with the stock FATE, their implementation using FATE shows an acceleration of 81 times and a reduction by 101 times of the traffic overhead.

Fang and Qian in 2021 [64] introduced a multi-party privacy-preserving machine learning framework called PFMLP (private federated multi-layer perceptron). This framework is based on partially homomorphic encryption and federated learning to protect privacy. The main objective is to mitigate membership inference attack. The authors proposed to counter such attack by hiding the shared gradients from the server using HE. In order to reduce the computational overhead of homomorphic encryption, they proposed to use an improved version of the Paillier scheme described by Jost et al. in 2015 [65]. Using this version, they speed up the training by 25–28% compared to the initial version of the Paillier scheme [30]. The authors conducted experimentation on MNIST and fatigue datasets and demonstrated that PFMLP achieves the same accuracy as the standard MLP (multi-layer perceptron) without HE.

Feng and Du in 2021 [66] proposed FLZip, a framework that uses gradients compression before encryption, to address the same challenges as BatchCrypt [63]. The key idea behind FLZip is to reduce the number of gradients to be encrypted by filtering insignificant gradients by introducing a hyperparameter. Then only the sparse significant gradients are encrypted. The lock in this scenario is how to design a feasible compression–encryption scheme that allows direct summation of ciphertexts without decryption. The authors focus on finding a "mergeable" compression scheme that maintains the addition property of HE. To achieve this, they proposed to select top-k significant gradients, encode them using key–value pairs, and then encrypt the values using the Paillier scheme [30]. Comparing their results to BatchCrypt, FLZip achieves a reduction in encryption and decryption operations by 6.4 times and 13.1 times, respectively, and shrinks the network footprints to and from the server by 5.9 times and 12.5 times, respectively, while maintaining model accuracy.

Liu et al. in 2022 [67] addressed the efficiency and the collusion threats in the previous works. For that, they developed a secure aggregation scheme, called doubly homomorphic secure aggregation (DHSA). The solution consists of two protocols: the Homomorphic Model Aggregation protocol (HMA) and the Masking Seed Agreement protocol (MSA). The HMA protocol utilizes a simple masking scheme based on a seed homomorphic random generator to hide the model updates. Then the demasking seed is securely calculated using the MSA protocol, which employs multi-key homomorphic encryption to ensure that the aggregation is only known by the clients. The work was compared to BatchCrypt [63] and the results show a speedup of up to 20 times while obtaining a similar accuracy to non-secure, uncompressed FedAvg.

Shin et al. [68] noticed that previous works do not protect the dataset size of each client. This information can inadvertently reveal sensitive data, such as the number of patients in the local hospital, rare diseases among the regions, etc. They considered a healthcare scenario and proposed a protocol for private federated averaging for the cross-silo setting using partial homomorphic encryption based on the Paillier scheme. In their protocol, each client interacts with a randomly selected neighbor to send the encrypted calculation result, instead of sending them to the server. The final result is then sent to the server for decryption. In this way, the local results of each client remain hidden from other clients and from the server.

Jin et al. in 2023 [69] proposed an HE–FL optimization scheme, named FedML-HE, that minimizes the size of model updates for encrypted computation while preserving privacy guarantees. The work addresses challenges related to communication and computation overhead (e.g., 10× reduction for HE-federated training of ResNet-50 and 40× reduction for BERT). In their approach, an honest-but-curious server aggregates the encrypted gradients from clients before decrypting them. Two techniques are introduced: parameter efficiency and parameter selection. In parameter efficiency, the goal is to reduce the model size through techniques such as model compression and parameter efficient tuning like in FLZip [66]. In parameter selection, the idea is to hide parts of the model instead of encrypting the whole model. The proposed solution was implemented using PALISADE for HE. The experimentation shows that the communication and computation overheads are reduced using the optimization techniques. The effectiveness of the parameter selection defense was also tested against gradient inversion, and the results show that encrypting 42% of the parameters is effective when using random selection mechanism, but using a more robust selection mechanism by selecting more important parameters is more efficient, and it is necessary to just encrypt 10% of the parameters to counter the DLG attack.

Table 4. Privacy-preserving FL using HE.

Ref	Year	Scheme	Key Idea	Trade-offs and Shortcomings
[63]	2020	Additive	Propose a batch additive scheme to reduce communication and computation overhead.	Batchcrypt is not applicable in Vertical FL.
[64]	2021	Additive	Hide shared gradients from from the server to protect against membership inference attack.	Scalability issue, computational and communication overhead
[66]	2021	Additive	Reduce the number of gradients to be encrypted by filtering insignificant gradients.	Scalability issues, computational and communication overhead
[67]	2022	Additive	Use a doubly homomorphic secure aggregation by using homomorphic encryption and masking technique.	Computational and communication overhead
[68]	2022	Additive	Additionally to previous work, protect the dataset size by adding interactions between clients using homomorphic encryption.	Computational and communication overhead
[69]	2023	Additive	Encrypting only a part of the model instead of the whole model. They showed that encrypting just 10% of the model parameter using a robust selection mechanism is efficient to counter DLG attack.	Need for theoretical analysis of the trade-offs among privacy guarantee, system overheads and model performance.

5.3. Discussion and Learned Lessons

The central challenge when using HE in FL is the computation and communication overhead. Unlike DP, which requires reducing the trade-off between privacy and model performance, in HE, the focus is on reducing the trade-off between privacy and computation overhead.

Several techniques have been explored to address this challenge, including batching, gradient compression, masking, parameter efficiency, and parameter selection. Batching techniques aim to encode many values within the same ciphertext while ensuring that the result can be obtained using only one operation on the ciphertext. Gradient compression, on the other hand, tries to compress the ciphertext to reduce the communication overhead. Masking is used as a lightweight technique that hides information using a mask seed, with the demasking seed calculated collaboratively using homomorphic encryption. Parameter efficiency and parameter selection techniques select only the efficient parameters and then encrypt only the most significant updates that may reveal much information about the data, rather than encrypting all the parameters. Previous works affirm that encrypting only the significant parameter is sufficient to counter privacy attacks.

Homomorphic encryption is well suited to counter eavesdropping attacks and the attacks that may exploit the updates coming from the client. It is also a good solution for anonymization, since the server will not access the updates provided by clients. However, the security of HE relies on the chosen scheme and the encryption keys. In addition, if the server accesses the final result, it still has the potential to perform a model inversion attack against the global model.

One drawback of HE is that the only operations possible are the addition and multiplication. Most research focus only on additive homomorphic encryption. Moreover, the computational complexity poses a challenging in terms of term efficiency and performance when applying HE in FL.

6. Combining DP and HE in Federated Learning

6.1. Related Works

Each technique has its own advantages and drawbacks in the context of privacy and security in federated learning. However, by combining these two techniques, we can potentially mitigate the drawbacks of each and achieve more comprehensive privacy protection. Several works have tried to combine these two techniques; Table 5 presents the different selected works, presenting the key ideas and the shortcomings.

Xu et al. in 2019 [70] proposed HybridAlpha, an FL framework that combines additive homomorphic encryption with differential privacy. The goal is to limit inference attacks from a curious aggregator during the process of learning and when using the final model. The system consists of a third-party authority (TPA) that generates the keys and distributes them, as well as an Inference Prevention Module. The module examines requests for private keys for specific vectors that may allow a specific curious aggregator to make an inference-enabling inner product. Hence, after receiving public keys from the TPA, the client will use LDP to protect their model updates from the server and then encrypt them. The server will then accomplish the aggregation before decrypting the data. The experimental results show that HybridAlpha can reduce the training time by 68% and data transfer volume by 92% while having similar privacy guarantees or model performance compared to existing works that use SMC, DP, and HE.

Wang et al. in 2020 [71] proposed two protocols to improve the utility of the data while guaranteeing better privacy. They proposed to build their solution based on the shuffler model proposed in Prochlo [72]. The challenge is to find a mechanism whose utility does not degrade with the evolution of the size of the data. They proposed a mechanism, named Shuffler-Optimal Local Hash (SOLH), and compared it to generalized random response (GRR) and unary encoding (RAPPOR). The results showed that SOLH outperformed GRR when the size of the data was large. However, when analyzing the security of this method, the authors found that collusion attacks may reveal information about the clients even when using DP. Therefore, they proposed a method called “Private Encrypted Oblivious Shuffle” that uses AHE to counter collusion attacks. The method was compared to various methods using shuffling, local hashing, and unary encoding.

Gu et al. in 2021 [73] proposed PRECAD, a framework for FL via crypto-aided differential privacy. This framework achieves differential privacy and uses cryptography against poisoning attacks. The author suggested using two non-colluding servers in an honest-but-curious model. The clients split their updates into two shares and send them to the servers. Additive secret sharing is used to verify the validity of the sharing, mitigating poisoning attacks. The servers then add CDP noise and conduct a secure aggregation step. The goal of this work is to improve the trade-off between privacy and robustness against poisoning attacks, contrary to previous works that try to improve the trade-off between privacy and utility. However, the experimentation also included tests on utility in order to validate the feasibility of the solution.

Sébert et al. in 2022 [74] published a work named “protecting data from all parties” that combines DP and HE in federated learning. In their work, each client applies successive transformations to achieve DP (clipping, noising, and quantization) then encrypt the data using HE before sending them to the server. HE protects the data from the semi-honest server, which performs calculations in an encrypted domain, while DP protects the data from the malicious clients. The challenges raised in this work are the computation cost of HE and the noise generation in DP. To decrease the computation cost, the authors suggest to use fixed-point numbers with a limited number of bits instead of floating-point numbers. They propose a new probabilistic quantization operator called "Poisson quantization" to handle the noise generation in a distributed manner, preventing the server from sharing the noise with other clients. In order to prove the feasibility of this framework, the experimentation was conducted using the FEMNIST dataset, a largely used dataset in previous works on federated learning.

One remarkable work that combines DP with HE is by Roy Chowdhurry et al. in 2020 [75]. The authors proposed crypt-ϵ a framework for executing DP programs. However, the framework is not specifically designed for the context of FL.

Table 5. Privacy preserving FL combining DP and HE

Ref	Year	Key Idea	Trade-offs and Shortcomings
[70]	2019	Add less noise by amplifying privacy by homomorphic encryption	Trade-off between privacy, communication, and computation.
[71]	2020	Amplify privacy with the shuffle model and protect data against collision attacks using Encrypted oblivious shuffle.	Increased system complexity.
[73]	2021	Split the updates into two shares and send them to two non-colluding servers that add CDP and use additive secret sharing to mitigate poisoning attacks and conduct secure aggregation.	Increased system complexity.
[74]	2022	Protect updates from the server using homomorphic encryption and protect global updates from clients using DP	Computational overhead.

6.2. Discussion and Learned Lessons

The combination of DP and HE in FL offers the potential to achieve a more comprehensive approach to privacy and security in federated learning. By leveraging the strengths of each technique, it becomes possible to mitigate their respective drawbacks and achieve enhanced privacy protection. HE can amplify the privacy offered by DP to protect the updates from all the parties, as in Sébert et al. [74]. While HE protects the intermediate updates from the server, DP also ensures the final model remains secure, preventing adversaries from performing model inversion attacks.

This combination is interesting also in terms of model performance. In fact, augmenting DP with HE can allow adding less noise and, by the way, having more utility of the data. The authors of the aforementioned work refer to this approach as crypto-aided differential privacy, emphasizing its potential for balancing between privacy and utility.

HE and DP can effectively mitigate various attacks from curious aggregators and from clients. By encrypting the data and applying differential privacy mechanisms, the privacy of the model updates and inference process can be safeguarded, preventing adversaries from extracting sensitive information. In addition, other attacks like collusion and poisoning attacks can be addressed using the combination of these techniques.

However, it is essential to acknowledge that the combination of DP and HE in FL does come with certain trade-offs and complexities. As the number of participants in the learning process increases, managing these entities can become challenging. Furthermore, the computational overhead associated with HE introduces resource consumption, impacting communication and computation within the system.

In summary, the integration of DP and HE in federated learning holds immense promise in enhancing privacy and security while striking a balance between utility and protection. However, it is crucial to carefully manage the system complexity and consider resource implications to fully harness the potential of this powerful privacy-preserving approach.

7. Discussion

While federated learning (FL) is often recognized as a technique that inherently protects privacy, it can still fall prey to numerous privacy attacks, as discussed in Section 3. The process of exchanging gradient updates across participating nodes in FL might inadvertently lead to potential privacy leaks. These leaks can expose sensitive aspects of the client's private data even without directly sharing the actual training datasets. This vulnerability is amplified due to the large number of participants involved in FL and the transparency of the framework's operations, which could provide ample opportunity for adversaries to launch powerful attacks.

In an effort to mitigate these vulnerabilities, our research highlights the potential of two techniques: differential privacy (DP) and homomorphic encryption (HE). In the academic community, DP is often split into three main categories: central differential privacy (CDP), local differential privacy (LDP), and the shuffle model. CDP is designed to shield raw data from potentially malicious clients, thus preventing unauthorized access. However, LDP goes a step further by also protecting data against adversarial servers. This additional layer of security, though, often comes at the expense of model performance due to the added noise.

This inherent trade-off gave rise to the exploration of the shuffle model, where privacy is fortified through a process of anonymization and shuffling. This technique severs the link between client-side updates and their origin, adding a further layer of privacy. Despite its advantages, the shuffle model requires trusting the shuffler as an 'honest-but-curious' server, which could be a potential point of vulnerability.

Balancing privacy and model performance is one of the major challenges when implementing DP. To ensure privacy, noise is added to the data, which can negatively impact the accuracy of the model. This inevitable trade-off is a critical consideration, prompting our exploration of other potential solutions, such as HE.

HE, though computationally expensive, has emerged as a promising technique. It promotes privacy by allowing only aggregated updates to be shared; thus, the aggregation server does not directly observe individual client updates. This approach minimizes accuracy loss, a crucial advantage. Yet, there are still concerns. For example, adversaries might potentially infer useful information from the final model using model inversion attacks. Further, the security provided by HE relies heavily on the strength of the encryption key and the security of the underlying encryption scheme. Unlike DP, it also does not offer plausible deniability, leaving users potentially exposed.

As outlined in Section 6, depending solely on one technique leaves potential gaps in security coverage. Therefore, an integrated approach, combining DP and HE, might offer a comprehensive solution. This hybrid model attempts to leverage the strengths of both DP and HE, offering accuracy from HE and plausible deniability from DP. However, this integration is far from straightforward. The challenge lies in navigating the trade-off between privacy, accuracy, and computational complexity to create a robust and efficient privacy-preserving FL framework.

In brief, the utilization of DP (differential privacy) and HE (homomorphic encryption) in federated learning can be depicted using Figures 2 and 3. The federated process, utilizing DP and HE, operates through a sequence of two alternating procedures, as depicted in Figures 2 and 3. The sequence kicks off with the server transmitting the global model to the clients. Subsequently, the clients proceed to train a local model and transmit their updates back to the server following the steps illustrated in Figure 2. After that, the server conducts secure aggregation and updates the global model based on the outlined process in Figure 3. These procedures persist until either convergence is reached or the maximum number of iterations is attained.

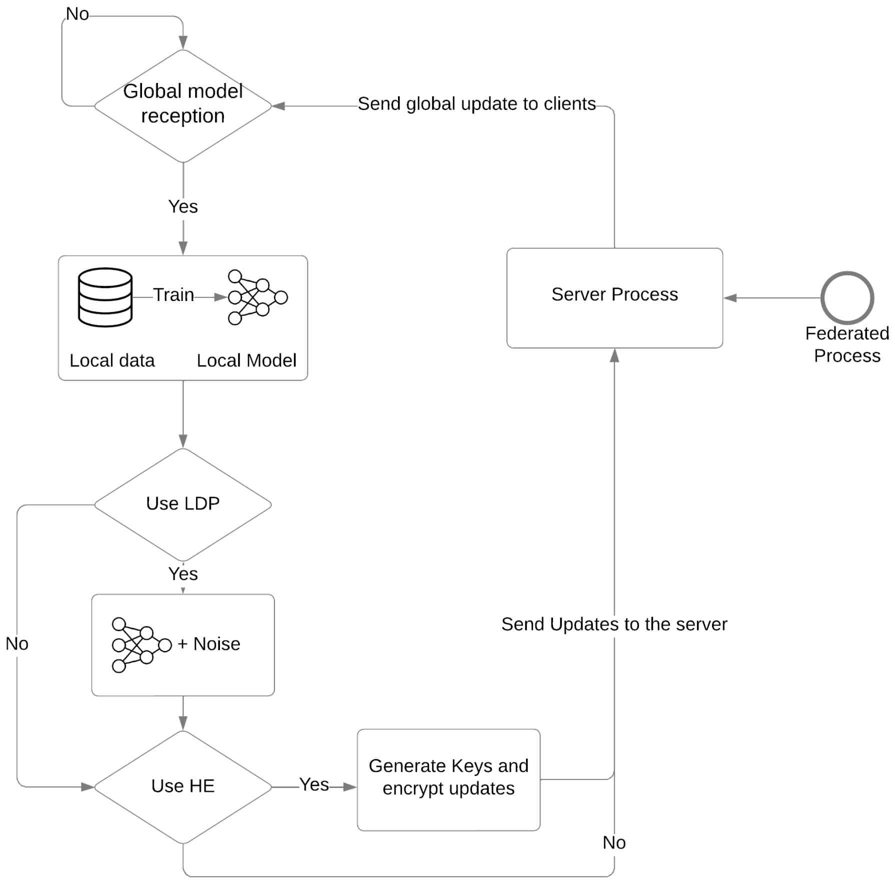

Figure 2. Client process in secure federated learning.

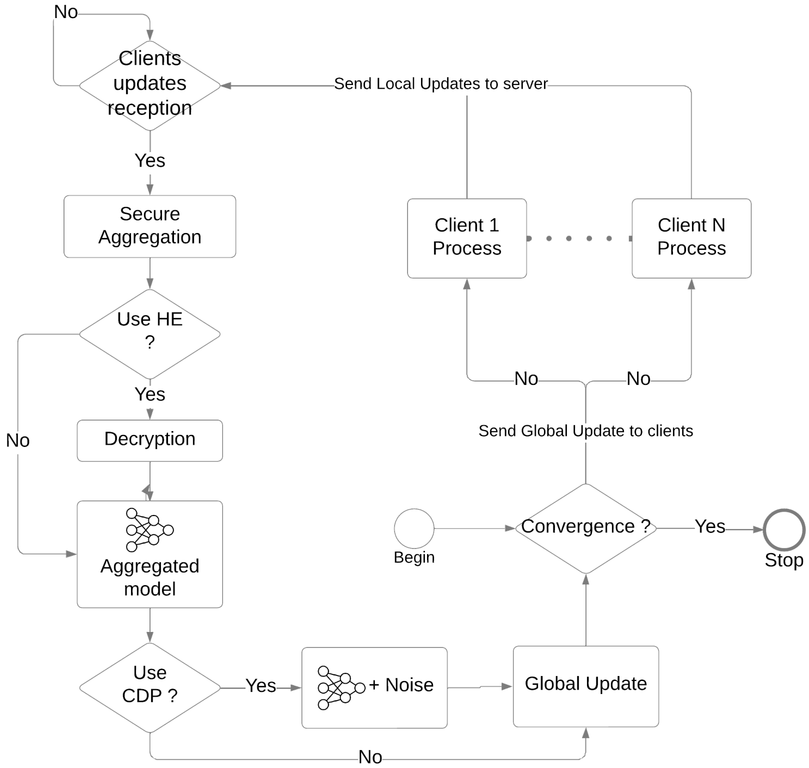

Figure 3. Server process in secure federated learning.

Ongoing Research

Inspired by recent seminal research on secured federated learning, specifically Time-varying Hierarchical Gradient Sparsification [76], we propose a novel homomorphic scheme to insert an additional layer for configuring an encryption mask. We conducted a preliminary overview analysis to determine the immediate impact of this proposed augmentation with HE in a secured federated environment.

It is feasible to reconfigure the encryption matrix for HE before applying the sparsification scheme. Assuming the idea of blind encryption, we propose configuring it through the Paillier modulus, while fetching a random value that is relatively prime to the multiplication modulo. Introducing the concept of relatively prime values can assist in distributing the masking of the matrix autonomously with less dependency on the operator. Similarly, to unmask the encryptor, we will evaluate the Paillier encryption with homomorphic computation. However, we anticipate that this logic may not fully mask the real value. Therefore, one alternative solution could be to generate two random values, ensuring that neither is 0. At the time of preparing this article, we have not investigated the deep-dive impact on the optimization of the double-layered secured matrix for the federated environment. This will be an extension of the present research.

8. Conclusions and Future Works

The core contributions of this study encompass a comprehensive analysis of recent implementations of DP and HE to handle privacy concerns within the context of FL. While FL is commonly perceived as a means of safeguarding privacy, our analysis has brought to light significant vulnerabilities present in various works. We delve into the spectrum of privacy attacks, illuminating their real-world relevance and implications for distributed learning. Furthermore, we offer nuanced insights into DP's deployment settings, HE's potential for safeguarding sensitive data, and the intersection of HE and DP techniques. Our work significantly augments the understanding of privacy strategies in FL and lays the groundwork for future advancements in this evolving landscape.

Regarding DP, the main challenge is striking a balance between privacy and accuracy. Addressing this challenge entails further research into devising more resilient mechanisms that introduce minimal noise while offering heightened privacy assurances. Furthermore, alternative relaxations of DP specifically designed for the FL environment or enhancing DP through auxiliary methods like anonymization, subsampling, or cryptography could offer novel avenues of investigation.

Concerning HE, the central challenge centers on mitigating the trade-off between privacy and computational complexity. Attacking this challenge requires a concerted effort to accelerate HE primitives while identifying algorithmic approaches to reduce the complexity of certain operations, such as division. By improving the efficiency of HE, we can simultaneously uphold privacy principles and mitigate computational overhead.

Furthermore, the combination of HE and DP is also an interesting direction. However, this amalgamation is far from straightforward, necessitating a careful equilibrium between computational complexity, model precision, and privacy considerations. As suggested in the work of Sébert et al. [74], combining these two techniques has the potential to safeguard raw data across all participants in the FL process, thereby showcasing a direction for future exploration.

Author Contributions: Conceptualization, R.A., S.B. (Soumya Banerjee) and S.B. (Samia Bouzefrane); methodology, R.A. and S.B. (Soumya Banerjee); validation, R.A., S.B. (Soumya Banerjee), S.B. (Samia Bouzefrane) and T.L.V.; investigation, R.A.; writing—original draft preparation, R.A., S.B. (Soumya Banerjee) and S.B. (Samia Bouzefrane); writing—review and editing, R.A. and T.L.V.; supervision, S.B. (Samia Bouzefrane). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by ROC team attached to CEDRIC Lab, Cnam Paris.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AHE	Additive Homomorphic Encryption
CDP	Centralized Differential Privacy
CNN	Convolutional Neural Network
DLG	Deep Leakage from Gradients
DNN	Deep Neural Network
DP	Differential Privacy
FC	Fully Connected
FCN	Fully Connected Network
FCNN	Fully connected neural network
FHE	Fully Homomorphic Encryption
FL	Federated Learning
GAN	Generative Adversial Network
GDPR	General Data Protection Regulation
GRNN	Generative Regression Neural Network
HE	Homomorphic Encryption
HFL	Horizontal Federated Learning
LDP	Local Differential Privacy
MIA	Membership Inference Attack
ML	Machine Learning
PCS	Prediction Confidence series
PHE	Partially Homomorphic Encryption
SGD	Stochastic Gradient Descent
SWHE	Somewhat Homomorphic Encryption
VFL	Vertical Federated Learning

References

1. Gartner. Gartner Identifies Top Five Trends in Privacy Through 2024. Available online: https://www.gartner.com/en/newsroom/press-releases/2022-05-31-gartner-identifies-top-five-trends-in-privacy-through-2024 (accessed on 1 June 2023).
2. European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). *Off. J. Eur. Union* **2016**, *4*, 1–88.
3. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
4. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
5. Zhao, B.; Mopuri, K.R.; Bilen, H. idlg: Improved deep leakage from gradients. *arXiv* **2020**, arXiv:2001.02610.
6. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting Gradients—How easy is it to break privacy in federated learning? In *Proceedings of the Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 16937–16947.
7. Ren, H.; Deng, J.; Xie, X. GRNN: Generative Regression Neural Network—A Data Leakage Attack for Federated Learning. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–24. [CrossRef]
8. Wei, W.; Liu, L.; Loper, M.; Chow, K.H.; Gursoy, M.E.; Truex, S.; Wu, Y. A Framework for Evaluating Client Privacy Leakages in Federated Learning. In *Proceedings of the Computer Security—ESORICS 2020*; Chen, L., Li, N., Liang, K., Schneider, S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 545–566.
9. Jin, X.; Chen, P.Y.; Hsu, C.Y.; Yu, C.M.; Chen, T. CAFE: Catastrophic Data Leakage in Vertical Federated Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 994–1006.

10. Zhang, J.; Zhang, J.; Chen, J.; Yu, S. GAN Enhanced Membership Inference: A Passive Local Attack in Federated Learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [CrossRef]
11. Mao, Y.; Zhu, X.; Zheng, W.; Yuan, D.; Ma, J. A Novel User Membership Leakage Attack in Collaborative Deep Learning. In Proceedings of the 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), Shaanxi, China, 23–25 October 2019; pp. 1–6. [CrossRef]
12. Chen, J.; Zhang, J.; Zhao, Y.; Han, H.; Zhu, K.; Chen, B. Beyond Model-Level Membership Privacy Leakage: An Adversarial Approach in Federated Learning. In Proceedings of the 2020 29th International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 3–6 August 2020; pp. 1–9. [CrossRef]
13. Wang, L.; Xu, S.; Wang, X.; Zhu, Q. Eavesdrop the composition proportion of training labels in federated learning. *arXiv* **2019**, arXiv:1910.06044.
14. Zhang, W.; Tople, S.; Ohrimenko, O. Leakage of Dataset Properties in Multi-Party Machine Learning. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Vancouver, BC, Canada, 11–13 August 2021; pp. 2687–2704.
15. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3347–3366. [CrossRef]
16. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *arXiv* **2021**. arXiv:1912.04977.
17. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
18. Lyu, L.; Yu, H.; Yang, Q. Threats to Federated Learning: A Survey. *arXiv* **2020**, arXiv:2003.02133.
19. Rodríguez-Barroso, N.; Jiménez-López, D.; Luzón, M.V.; Herrera, F.; Martínez-Cámara, E. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Inf. Fusion* **2023**, *90*, 148–173. [CrossRef]
20. Zhang, K.; Song, X.; Zhang, C.; Yu, S. Challenges and future directions of secure federated learning: A survey. *Front. Comput. Sci.* **2021**, *16*, 165817. [CrossRef]
21. Li, Z.; Sharma, V.; Mohanty, S.P. Preserving Data Privacy via Federated Learning: Challenges and Solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 8–16. [CrossRef]
22. Yin, X.; Zhu, Y.; Hu, J. A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions. *ACM Comput. Surv.* **2021**, *54*, 1–36. [CrossRef]
23. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
24. Kaissis, G.A.; Makowski, M.R.; Rückert, D.; Braren, R.F. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat. Mach. Intell.* **2020**, *2*, 305–311. [CrossRef]
25. Gu, X.; Sabrina, F.; Fan, Z.; Sohail, S. A Review of Privacy Enhancement Methods for Federated Learning in Healthcare Systems. *Int. J. Environ. Res. Public Health* **2023**, *20*, 6539. [CrossRef]
26. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
27. Niknam, S.; Dhillon, H.S.; Reed, J.H. Federated Learning for Wireless Communications: Motivation, Opportunities and Challenges. *arXiv* **2020**, arXiv:1908.06847.
28. Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; Naor, M. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2006*; Vaudenay, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 486–503.
29. Albrecht, M.; Chase, M.; Chen, H.; Ding, J.; Goldwasser, S.; Gorbunov, S.; Halevi, S.; Hoffstein, J.; Laine, K.; Lauter, K.; et al. *Homomorphic Encryption Security Standard*; Technical Report; HomomorphicEncryption.org: Toronto, ON, Canada, 2018.
30. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; pp. 223–238.
31. Gentry, C. Fully Homomorphic Encryption Using Ideal Lattices. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 31 May–2 June 2009; Association for Computing Machinery: New York, NY, USA; pp. 169–178. [CrossRef]
32. Liu, P.; Xu, X.; Wang, W. Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives. *Cybersecurity* **2022**, *5*, 4. [CrossRef]
33. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks against Machine Learning Models. *arXiv* **2017**, arXiv:1610.05820.
34. Salem, A.; Zhang, Y.; Humbert, M.; Berrang, P.; Fritz, M.; Backes, M. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In Proceedings of the 2019 Network and Distributed System Security Symposium, San Diego, CA, USA, 24–27 February 2019. [CrossRef]
35. Pustozerova, A.; Mayer, R. Information Leaks in Federated Learning. In Proceedings of the 2020 Workshop on Decentralized IoT Systems and Security, San Diego, CA, USA, 23 February 2020. [CrossRef]
36. Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Zhang, X. Source Inference Attacks in Federated Learning. *arXiv* **2021**, arXiv:2109.05659.

37. Nasr, M.; Shokri, R.; Houmansadr, A. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), Francisco, CA, USA, 20–22 May 2019; pp. 739–753. [CrossRef]
38. Gu, Y.; Bai, Y.; Xu, S. CS-MIA: Membership inference attack based on prediction confidence series in federated learning. *J. Inf. Secur. Appl.* **2022**, *67*, 103201. [CrossRef]
39. Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; Qi, H. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2512–2520. [CrossRef]
40. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1322–1333. [CrossRef]
41. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; CCS '17, pp. 603–618. [CrossRef]
42. Ateniese, G.; Mancini, L.V.; Spognardi, A.; Villani, A.; Vitali, D.; Felici, G. Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers. *Int. J. Secur. Netw.* **2015**, *10*, 137–150. [CrossRef]
43. Melis, L.; Song, C.; Cristofaro, E.D.; Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. *arXiv* **2018**, arXiv:cs.CR/1805.04049.
44. Ganju, K.; Wang, Q.; Yang, W.; Gunter, C.A.; Borisov, N. Property Inference Attacks on Fully Connected Neural Networks Using Permutation Invariant Representations. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 15–19 October 2018; Association for Computing Machinery: New York, NY, USA; pp. 619–633. [CrossRef]
45. Zhou, J.; Chen, Y.; Shen, C.; Zhang, Y. Property Inference Attacks Against GANs. *arXiv* **2021**, arXiv:2111.07608.
46. Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J.M.; Kautz, J.; Molchanov, P. See through Gradients: Image Batch Recovery via GradInversion. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 16332–16341. [CrossRef]
47. McMahan, H.B.; Ramage, D.; Talwar, K.; Zhang, L. Learning Differentially Private Language Models Without Losing Accuracy. *arXiv* **2017**, arXiv:1710.06963.
48. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
49. Geyer, R.C.; Klein, T.; Nabi, M. Differentially Private Federated Learning: A Client Level Perspective. *arXiv* **2017**, arXiv:1712.07557.
50. Choudhury, O.; Gkoulalas-Divanis, A.; Salonidis, T.; Sylla, I.; Park, Y.; Hsu, G.; Das, A. Differential Privacy-enabled Federated Learning for Sensitive Health Data. *arXiv* **2019**, arXiv:1910.02578.
51. Hu, R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized Federated Learning With Differential Privacy. *IEEE Internet Things J.* **2020**, *7*, 9530–9539. [CrossRef]
52. Bhowmick, A.; Duchi, J.; Freudiger, J.; Kapoor, G.; Rogers, R. Protection Against Reconstruction and Its Applications in Private Federated Learning. *arXiv* **2019**, arXiv:1812.00984.
53. Liu, R.; Cao, Y.; Yoshikawa, M.; Chen, H. FedSel: Federated SGD under Local Differential Privacy with Top-k Dimension Selection. *arXiv* **2020**, arXiv:2003.10637.
54. Ni, L.; Huang, P.; Wei, Y.; Shu, M.; Zhang, J. Federated Learning Model with Adaptive Differential Privacy Protection in Medical IoT. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8967819. [CrossRef]
55. Sun, L.; Qian, J.; Chen, X. LDP-FL: Practical Private Aggregation in Federated Learning with Local Differential Privacy. *arXiv* **2021**, arXiv:2007.15789.
56. Chamikara, M.A.P.; Liu, D.; Camtepe, S.; Nepal, S.; Grobler, M.; Bertok, P.; Khalil, I. Local Differential Privacy for Federated Learning. *arXiv* **2022**, arXiv:2202.06053.
57. Shen, X.; Jiang, H.; Chen, Y.; Wang, B.; Gao, L. PLDP-FL: Federated Learning with Personalized Local Differential Privacy. *Entropy* **2023**, *25*, 485. [CrossRef] [PubMed]
58. Girgis, A.; Data, D.; Diggavi, S. Renyi Differential Privacy of The Subsampled Shuffle Model In Distributed Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 29181–29192.
59. Girgis, A.M.; Data, D.; Diggavi, S. Differentially Private Federated Learning with Shuffling and Client Self-Sampling. In Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, 12–20 July 2021; pp. 338–343. [CrossRef]
60. Girgis, A.; Data, D.; Diggavi, S.; Kairouz, P.; Suresh, A.T. Shuffled Model of Differential Privacy in Federated Learning. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 13–15 April 2021; pp. 2521–2529.

61. Li, Y.; Chang, T.H.; Chi, C.Y. Secure Federated Averaging Algorithm with Differential Privacy. In Proceedings of the 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), Espoo, Finland, 21–24 September 2020; pp. 1–6. [CrossRef]
62. Yaldiz, D.N.; Zhang, T.; Avestimehr, S. Secure Federated Learning against Model Poisoning Attacks via Client Filtering. *arXiv* **2023**, arXiv:2304.00160.
63. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference, Boston, MA, USA, 15–17 July 2020; USENIX ATC'20.
64. Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. *Future Internet* **2021**, *13*, 94. [CrossRef]
65. Jost, C.; Lam, H.; Maximov, A.; Smeets, B.J.M. Encryption Performance Improvements of the Paillier Cryptosystem. *IACR Cryptol. ePrint Arch.* **2015**, 864. Available online: https://eprint.iacr.org/2015/864 (accessed on 2 June 2023).
66. Feng, X.; Du, H. FLZip: An Efficient and Privacy-Preserving Framework for Cross-Silo Federated Learning. In Proceedings of the 2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Melbourne, Australia, 6–8 December 2021; pp. 209–216. [CrossRef]
67. Liu, Z.; Chen, S.; Ye, J.; Fan, J.; Li, H.; Li, X. DHSA: Efficient doubly homomorphic secure aggregation for cross-silo federated learning. *J. Supercomput.* **2023**, *79*, 2819–2849. [CrossRef]
68. Shin, Y.A.; Noh, G.; Jeong, I.R.; Chun, J.Y. Securing a Local Training Dataset Size in Federated Learning. *IEEE Access* **2022**, *10*, 104135–104143. [CrossRef]
69. Jin, W.; Yao, Y.; Han, S.; Joe-Wong, C.; Ravi, S.; Avestimehr, S.; He, C. FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System. *arXiv* **2023**, arXiv:2303.10837.
70. Xu, R.; Baracaldo, N.; Zhou, Y.; Anwar, A.; Ludwig, H. HybridAlpha: An Efficient Approach for Privacy-Preserving Federated Learning. In Proceedings of the Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. *arXiv* **2019**, arXiv:1912.05897.
71. Wang, T.; Ding, B.; Xu, M.; Huang, Z.; Hong, C.; Zhou, J.; Li, N.; Jha, S. Improving Utility and Security of the Shuffler-Based Differential Privacy. *Proc. VLDB Endow.* **2020**, *13*, 3545–3558. [CrossRef]
72. Bittau, A.; Erlingsson, U.; Maniatis, P.; Mironov, I.; Raghunathan, A.; Lie, D.; Rudominer, M.; Kode, U.; Tinnes, J.; Seefeld, B. Prochlo: Strong Privacy for Analytics in the Crowd. In Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, 28–31 October 2017; SOSP '17, pp. 441–459. [CrossRef]
73. Gu, X.; Li, M.; Xiong, L. PRECAD: Privacy-Preserving and Robust Federated Learning via Crypto-Aided Differential Privacy. *arXiv* **2021**, arXiv:2110.11578.
74. Sébert, A.G.; Sirdey, R.; Stan, O.; Gouy-Pailler, C. Protecting Data from all Parties: Combining FHE and DP in Federated Learning. *arXiv* **2022**, arXiv:2205.04330.
75. Roy Chowdhury, A.; Wang, C.; He, X.; Machanavajjhala, A.; Jha, S. Crypt ϵ: Crypto-Assisted Differential Privacy on Untrusted Servers. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 603–619. [CrossRef]
76. Liu, T.; Wang, Z.; He, H.; Shi, W.; Lin, L.; An, R.; Li, C. Efficient and Secure Federated Learning for Financial Applications. *Appl. Sci.* **2023**, *13*, 5877. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.