Privacy-Preserving Aggregation in Federated Learning: A Survey

Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, *Senior Member, IEEE,* and Jun Zhao, *Member, IEEE*

Abstract—Over the recent years, with the increasing adoption of Federated Learning (FL) algorithms and growing concerns over personal data privacy, Privacy-Preserving Federated Learning (PPFL) has attracted tremendous attention from both academia and industry. Practical PPFL typically allows multiple participants to individually train their machine learning models, which are then aggregated to construct a global model in a privacy-preserving manner. As such, Privacy-Preserving Aggregation (PPAgg) as the key protocol in PPFL has received substantial research interest. This survey aims to fill the gap between a large number of studies on PPFL, where PPAgg is adopted to provide a privacy guarantee, and the lack of a comprehensive survey on the PPAgg protocols applied in FL systems. This survey reviews the PPAgg protocols proposed to address privacy and security issues in FL systems. The focus is placed on the construction of PPAgg protocols with an extensive analysis of the advantages and disadvantages of these selected PPAgg protocols and solutions. Additionally, we discuss the open-source FL frameworks that support PPAgg. Finally, we highlight significant challenges and future research directions for applying PPAgg to FL systems and the combination of PPAgg with other technologies for further security improvement.

1 Introduction

Over the recent years, with the increasing adoption of machine learning (ML) algorithms and growing concern of data privacy, the scenario where different data owners, e.g., mobile devices or cloud servers, jointly solve a machine learning problem, i.e., train an ML model, while preserving their data privacy has attracted tremendous attention from both academia and industry. In this connection, federated learning (FL) [1] is proposed to achieve privacy-enhanced distributed machine learning schemes, and has been applied to a wide range of scenarios such as Internet of Things (IoT) [2], [3], [4], [5], [6], healthcare [7], [8], [9], [10], computer vision [11], [12], [13], and recommendation [14], [15]. A standard FL system typically enables different participants, i.e., data owners, to individually train an ML model using their local data, which are then aggregated by a central server to construct a global FL model. However, as pointed out in [16], with only a small portion of the user's model, an attacker, e.g., a malicious central server, can easily reconstruct the user's data with pixel-wise accuracy for images and token-wise matching for texts. To mitigate such so-called "deep leakage from gradients", Privacy-Preserving Technique (PPT) such as Homomorphic Encryption (HE) [17], Multi-Party Computation (MPC) [18], Differential Privacy (DP) [19], and infrastructures such as blockchain [20] and Trusted Execution Environment (TEE) [21] have been proposed to enhance FL systems by aggregating the users' locally trained models in a privacy-preserving manner. As such, Privacy-Preserving Aggregation (PPAgg) as the key protocol in Privacy-Preserving Federated Learning (PPFL) has received substantial research interest.

In general, one can enhance PPFL by constructing PPAgg protocols widely adopted in standard distributed Privacy-Preserving Machine Learning (PPML). However, compared to PPML, PPFL further considers heterogeneous participants of, e.g., different computational power and bandwidth, and more complicated threat models regarding privacy and security [22]. For example, in a cross-device FL setting, participants are usually resource-constrained mobile devices that may drop out of the system at any time (see Section 2 for more details). This requires the PPAgg protocol to provide both cost-effective execution and dropout resilience. Meanwhile, the security and privacy issues in PPFL systems may come from insiders, e.g., FL participants, or outsiders, e.g., simulated dummy participants, from a single adversary, e.g., the central server, or multiple adversaries, e.g., several colluding participants. Besides, adversaries can be considered to be semi-honest, i.e., try to learn the private information of honest participants without deviating from the FL protocol, or active malicious, i.e., try to learn the private information of other honest participants by deviating arbitrarily from the FL protocol, e.g., by manipulating messages. Therefore, specific designs of PPAgg protocols are required to achieve PPFL in different application scenarios. We should note that although the PPAgg protocol aims only at computing the sum of vectors from different participants, it may become the bottleneck for large-scale FL systems, especially when complicated constructions are adopted to provide many security and privacy guarantees.

- *Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, 50 Nanyang Ave, 639798.*
E-mail: {ziyao002, jiale001, wenzhuo001, jiani001} @e.ntu.edu.sg, {kwokyan.lam, junzhao} @ntu.edu.sg. Corresponding author: Jun Zhao

This research/project is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. This paper is also supported in part by the Nanyang Technological University (NTU) Startup Grant, the NTU-WASP Joint Project, and the Singapore Ministry of Education Academic Research Fund under Tier 1 Grant RG24/20, Tier 1 Grant RG97/20, and Tier 2 Grant MOE2019-T2-1-176.
Manuscript received April 4, 2022; revised May 20, 2022 and July 1, 2022; accepted July 6, 2022. Date of publication TBD, 2022.

Comparison with other surveys. Currently, few existing surveys on PPFL perceive the construction and organization of PPFL from the perspective of privacy-preserving aggregation protocols. In particular, the surveys [1], [22], [23] give a comprehensive introduction to federated learning. The surveys in [24], [25] extensively analyze the privacy and security threats to FL systems with discussions on possible attacks and defenses. The survey in [26], [27] presents the PPFL applications to the Internet of Things, and the surveys in [28], [29] discuss the integration of PPFL and edge computing. Several research papers such as [30], [31], [32] have surveyed some PPAgg protocols in FL. However, they do not provide extensive discussion regarding different constructions and threat models. To the best of our knowledge, there is no survey specifically discussing the aggregation protocol, as a key privacy-preserving technique adopted in PPFL systems. This motivates us to deliver the survey with a comprehensive literature review on the construction of PPAgg protocols in PPFL with a discussion on their application scenarios. We note that many studies focus on optimizing the performance and efficiency of standard FL training. However, this survey concentrates on FL systems from a privacy and security perspective, hence they are out of the scope of this paper, and interested readers can refer to [22], [33], [34], [35] for the state-of-the-art FL training algorithms. For convenience, the related works in this survey are classified based on their main technique used to guarantee privacy, as one PPAgg protocol may involve several supported privacy-preserving techniques to provide different properties. For example, SecAgg [36], which adopts both masking and secret sharing technique, is classified as a masking-based aggregation in this survey since the masking technique is deployed to protect the users' model privacy while secret sharing mainly provides the dropout-resilience. These major classifications consist of (i) masking-based aggregation, (ii) HE-based aggregation, (iii) MPC-based aggregation, (iv) DP-based aggregation, (v) blockchain-based aggregation, and (vi) TEE-based aggregation.

Organisation of the paper. The rest of this paper is organized as follows. Section 2 describes the general architecture of and privacy threats to federated learning systems. Section 3 presents the fundamentals of supporting tools that are commonly used for privacy-preserving aggregation. Section 4 reviews different constructions of PPAgg protocols in federated learning, followed by the discussions on open-source FL frameworks that support PPAgg in Section 5. Section 6 outlines challenges and future research directions. Section 7 summarizes and concludes the paper.

2 Overview and Fundamentals of Federated Learning

This section will give an overview of the federated learning on its concepts, data organization, working mechanism, and privacy threats to FL systems.

2.1 Overview of Federated Learning

A federated learning scheme typically enables different participants, i.e., data owners, to individually train an ML model using their local data, which are then aggregated with the coordination of a central server to construct a global FL model. The FL participants can be divided into two classes, i.e., (i) a set of n users $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ that each user $u_i \in \mathcal{U}$ has a local dataset $\mathcal{D}_i$, and (ii) a central server S.

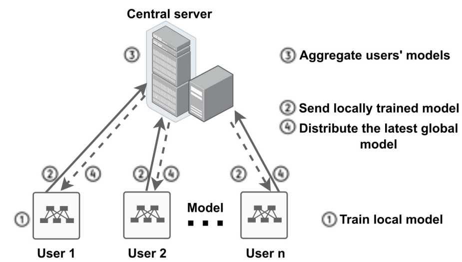


Fig. 1. A typical workflow of the training process in FL systems.

As shown in Figure 1, a typical FL scheme works by repeating the following steps until training is stopped. (i) Local model training: each FL user u_i trains its model $\mathcal{M}_i$ using the local dataset $\mathcal{D}_i$. (ii) Model uploading: each FL user u_i uploads its locally trained model $\mathcal{M}_i$ to the central server S. (iii) Model aggregation: the server S collects and aggregates users' models to update the global model $\mathcal{M}$. (iv) Model updating: the server S updates the global model $\mathcal{M}$ and distributes it to all FL users, based on which the users train their new local models in a new FL round.

Furthermore, as mentioned earlier, FL systems usually involve heterogeneity with respect to the data format, computational power, bandwidth, etc. Therefore, specific designs of PPAgg protocols are required to achieve PPFL in different settings. Here we keep the consistency of the classification of FL settings from [22], i.e., Cross-Silo setting and Cross-Device setting, as described in Table 1. In general, a cross-silo setting focuses on the FL systems that consist of only reliable participants, i.e., several companies and governments, with sufficient communication bandwidth and computational power. In contrast, a cross-device setting aims to provide FL solutions to many resource-constrained unreliable participants, such as IoT devices and mobile devices.

TABLE 1
An adaptive classification of FL settings from [22].

	Cross-Silo	Cross-Device
Participants	Several different organizations	A large number of mobile or IoT devices
Distribution scale	Typically 2-100 users	Up to 10^{10} users
Primary bottleneck	Computation or communication	Communication due to slow connections
User reliability	Relatively few failures	Highly unreliable
User statefulness	Online from round to round	May drop out at anytime

2.2 Privacy Threats to Federated Learning

As discussed earlier, privacy threats in FL systems can lead to different information leakage and come from both insiders, e.g., FL participants, and outsiders, e.g., eavesdroppers. However, the capability of insider adversaries is generally stronger than that of outsider adversaries, as one can adopt cryptographic tools to, e.g., achieve secure communication channels and verify the identities of outsiders, to mitigate the privacy issue caused by outsider attacks. Therefore, our discussion of privacy threats in FL will focus primarily on insider privacy leakages. We should note that in this survey, we focus on PPAgg protocols that protect the privacy of honest FL participants' inputs. The protocols that aim to guarantee security against non-privacy attacks, e.g., backdoor attacks or poisoning attacks [37], fall into Byzantine-robust aggregation [38], [39], [40] which are out of the scope of this paper. However, in Section 4.7, we will discuss the potential integration of them with PPAgg protocols for further security improvement. The privacy threats in FL can be categorized into the following two general forms.

- Privacy threats to users' models: it is proved that local data of an individual FL participant could be revealed through a small portion of its locally trained model [16], which directly breaks the basic privacy guarantee of standard federated learning. Therefore, a large number of PPAgg protocols focus on dealing with such privacy leakage.
- Privacy threats to the global model: standard FL schemes assume that the global models are over plaintext. However, the privacy of the global models is considered in some FL application scenarios. Therefore, PPAgg protocols are required to provide the privacy guarantee of both users' models and the global model.

Furthermore, privacy leakages may come from a single adversary or multiple adversaries, which can take one of the following two general forms.

- Single adversary: a single, non-colluding participant, which can be an FL user, the central server, or the third party. Note that the central server usually has a stronger capability than a single FL user.
- Colluding adversaries: collusion may happen with or without the central server. Note that colluding FL users with the central server and more adversaries usually lead to a greater risk of privacy leakages.

Last but not least, the capability of adversaries should be taken into account, which can be classified into the following three general forms.

- Honest: follow the protocol honestly.
- Passive malicious (honest-but-curious or semi-honest): try to learn the private information of honest participants without deviating from the protocol.
- Active malicious (malicious when the context is clear): can deviate from the protocol at any time, e.g., manipulating identity or sending fraudulent messages to others.

3 Techniques for Privacy-Preserving Aggregation

In this section, we give an overview of the supporting tools to construct privacy-preserving aggregation protocols in FL systems, including some privacy-preserving techniques such as one-time pad, homomorphic encryption, secure multi-party computation, differential privacy, and infrastructures such as blockchain and trusted execution environment.

3.1 One-time Pad

In cryptography, a trusted authority encrypts the message with a One-Time Pad (OTP) and sends OTP to the user's trusted device, where the user can decrypt and obtain the original message. In such a way, the randomness of OTP guarantees that each encryption of message is unique and has no relation to any other encryption, and thus there is no way to break the messages encrypted by OTP [41].

Specifically, OTP can be adopted to encrypt a message in an additive or multiplicative manner. For example, by adding random generated OTP r to a message x in a finite field $\mathbb{F}_p$ to obtain the encrypted message y, i.e., $y = x + r \bmod p$, the message x is perfectly masked by r. Similarly, the multiplicative masking by OTP has the same security guarantee as that of additive masking, provided the message $x \neq 0$, i.e., $y = x \cdot r \bmod p$. In this way, FL participants can mask their models to preserve their privacy. However, keeping the correctness of aggregation on the masked model is not a straightforward task. Therefore, well-designed masking techniques with aggregation protocols are proposed to cancel the masks to obtain the correct results. We will review those related works in Section 4.1.

3.2 Secure Multi-Party Computation

Secure Multi-Party Computation (MPC or SMPC) broadly encompasses all cryptographic techniques for privacy-preserving function evaluations between multiple parties, including but not limited to homomorphic encryption (HE), Garbled Circuit (GC) [42], Oblivious Transfer (OT) [43], GMW [44] and Secret Sharing Scheme (SSS). Since its general definition in [18], MPC has moved from pure theoretical interests to practical implementations, and has developed many generic frameworks to support secure computation in two-party, e.g., ABY [45], and in multi-party settings, e.g., SPDZ family [46] and ABY3 [47]. Besides, with the development of machine learning in recent years, an efficient MPC scheme supporting privacy-preserving machine learning has attracted tremendous attention from both academia and industry, e.g., [45], [47], [48], [49], [50], [51]. Note that these generic PPML schemes can be straightforwardly extended to achieve PPFL systems where users share their local data or locally trained models to several participants, e.g., non-colluding servers, that keep online during the whole protocol execution (see Section 4.3). Since the applications of pure-MPC to FL usually lead to impractical communication overheads, especially for large-scale FL systems with complicated ML models, simple secret sharing schemes are always considered to be integrated with other PPTs to achieve privacy-preserving aggregation or training in FL.

Secret sharing refers to a cryptographic primitive that allows a secret to be distributed and reconstructed among a set of participants. More formally, a (t, n) threshold secret sharing scheme allows one to distribute a secret s to n parties $p_1, p_2, \ldots, p_n$ such that only a subset of these parties of which the number is not less than the threshold t can reconstruct the secret s, while any subset of parties of which the number is less than the threshold t does not obtain any information about the secret s. In specific, additive secret sharing and Shamir secret sharing are two widely-used schemes to construct such MPC protocols. A secret sharing is linear or additive if the reconstruction of the secret from the shares is a linear mapping or additive homomorphic. For example, in an additive secret sharing scheme, the secret s is divided into n pieces $s_1, s_2, \ldots, s_n$ over a finite field $\mathbb{F}_p$ such that $s = \sum_{i=1}^{n} s_n \bmod p$. Such additive secret sharing is the basic structure of many generic MPC protocols such as SPDZ [46]. Unlike additive secret sharing, Shamir secret sharing leverages non-linear mapping to reconstruct the secret. Specifically, for a (t, n) Shamir scheme, to share a secret s, one needs to randomly select $t-1$ elements $a_1, a_2, \ldots, a_{t-1}$ from a finite field $\mathbb{F}_p$ and let $a_0 = s$, to construct the polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \bmod p$$

Then one can n distinct points on the curve defined by the Lagrange polynomial except for the point $(0, s)$ and distribute them as shares to n parties. To reconstruct the secret s, once one has collected at least t Shamir shares (x_i, y_i), the constant term of the above Lagrange polynomial can be obtained by calculating

$$s = f(0) = \sum_{j=0}^{t-1} y_j \prod_{m=0, m \neq j}^{t-1} \frac{x_m}{x_m - x_j}$$

We should note that although the Shamir scheme involves non-linear mapping to reconstruct the secret, operations on its shares still hold additive homomorphism. Besides, the threshold structure of Shamir schemes makes it natural to be used to handle dropped users and to construct verification protocols.

3.3 Homomorphic Encryption

Homomorphic Encryption (HE) is a kind of encryption scheme that allows one to perform function evaluations over encrypted data while preserving the function features and data format. As an example of additive public-key HE scheme with the key pair (pk, sk), for two messages m_1 and m_2, one can compute $Enc(m_1 + m_2, pk)$ using $Enc(m_1, pk)$ and $Enc(m_2, pk)$ without knowing any information about m_1 and m_2, where $Enc(\cdot)$ denotes the encryption function and pk is the public key. After that, one can obtain $m_1 + m_2$ relying on the corresponding decryption function $Dec(\cdot)$ and the secret key sk. Note that for simplicity, we sometimes abuse the notation $Enc(\cdot)$ and $Dec(\cdot)$ without using pk and sk when the context is clear.

In general, HE schemes can be categorized according to the number of allowed arithmetic operations on the encrypted data as follows.

- Partially Homomorphic Encryption (PHE): allows an unlimited number of operations but with only one type, e.g., addition or multiplication.
- Somewhat Homomorphic Encryption (SWHE): allows some types of operations but with a limited number of times, e.g., one multiplication with an unlimited number of additions.
- Fully Homomorphic Encryption (FHE): allows unlimited types of arithmetic operations with an unlimited number of times.

For privacy-preserving aggregation in FL systems, as it involves only one type of arithmetic operation, i.e., addition, the PHE scheme becomes the natural option. For example, Paillier crypto-systems [52] are widely adopted in FL to enable addition over encrypted data, hence protecting users' privacy. ElGamal crypto-systems [53] can also be adapted by converting the aggregation to the product. Furthermore, to protect the privacy of the whole FL workflow, e.g., the global model, FL users need to train their local model based on an encrypted global model, which requires complicated function evaluation over ciphertext. Therefore, FHE schemes are considered to be the only choice. Otherwise, one must convert all encrypted data to a secretly shared format and leverage MPC protocols to train the ML model. Among FHE schemes, lattice-based CKKS [54], [55] is the most popular scheme used in privacy-preserving FL due to the good trade-off for their efficiency and accuracy.

Note that PHE schemes, e.g., Paillier and ElGamal, are usually more efficient than SWHE, while SWHE schemes, e.g., leveled BGV, are usually more efficient than FHE, e.g., BGV and CKKS. Due to the privacy requirements of PPFL systems, SWHE outperforms neither PHE for privacy-preserving aggregation nor FHE for privacy-preserving training. Thus, SWHE is often considered to be the underlying tool to support other high-level cryptographic protocols. For example, leveled BGV (a type of SWHE) is used to construct the generic MPC protocol SPDZ [56]. Besides, we should note that by sacrificing some efficiency, all HE schemes can be extended to their threshold or multi-key version, e.g., threshold Paillier [57] and multi-key CKKS [58]. In this case, the secret key is distributed among all participants that are involved in the key generation process. Hence, one must corrupt more FL participants to break the security compared to those of standard HE schemes.

3.4 Differential Privacy

Differential Privacy (DP) is a technique that gives a solution to the paradox of learning knowledge from a large dataset but securing the privacy of individual participants [19]. Referring to descriptions in [59], [60], the definition of DP can be summarized as: A randomized mechanism $\mathcal{M}$ is differentially private if for any two neighboring databases (ND) $\mathcal{X}$ and $\mathcal{X}'$, and for all possible outputs, it satisfies ϵ-DP when

$$\frac{P[\mathcal{M}(\mathcal{X}) \in S]}{P[\mathcal{M}(\mathcal{X}') \in S]} \leq \exp(\epsilon). \tag{1}$$

Here, $P[\cdot]$ represents the probability, ϵ is the privacy budget that controls the different degrees of the two outputs from $\mathcal{M}$ with the two ND as inputs. The smaller the ϵ is, the higher the participant's privacy level and the lower the utility. Databases are ND when they follow any of the two conditions: (i) if $\mathcal{X}$ and $\mathcal{X}'$ are two datasets that have at most one record different; (ii) if $\mathcal{X}$ and $\mathcal{X}'$ have one entry different.

The function sensitivity $S_{\mathcal{M}}$ of a randomized mechanism $\mathcal{M}$ can be represented as

$$S_{\mathcal{M}} = \max_{\mathcal{X}, \mathcal{X}'} \left\| \mathcal{M}(\mathcal{X}) - \mathcal{M}(\mathcal{X}') \right\|_p ,$$

which measures the maximum difference of the outputs when input a pair of ND. Here the ℓ_p sensitivity can be calculated by 1-norm, 2-norm, or other distance calculation methods based on different setting requirements.

In FL, the information privacy in a pair of ND $\mathcal{X}$, $\mathcal{X}'$ can be protected by adding random noise to the data or different model parameters based on a selected differentially private-mechanism $\mathcal{M}$ [60], [61]. Some commonly used noise generation mechanisms include the Gaussian mechanism [62] and Laplace mechanism [60].

There are different definitions for central/global differential privacy (GDP) and local differential privacy (LDP) concepts. To make these two concepts more explicit and easier to understand, we distinguish them in this paper from the perspective of who adds noise to the FL training parameters. The approaches that perturb the parameters by the global server or a trusted third-party server that ensure the privacy of the global model are regarded as GDP-based works. Techniques that add noise by local users or aim to protect the local models' privacy are treated as LDP-based methods. GDP follows Eq.(1). LDP is held by

$$\frac{P[\mathcal{M}_n(\mathcal{X}_n) \in S_n]}{P[\mathcal{M}_n(\mathcal{X}_n') \in S_n]} \leq \exp(\epsilon_n)$$

for any $\mathcal{X}_n$ and $\mathcal{X}_n'$, where n means the n-th participant. The nature of differential privacy can promise individual-indistinguishable, which makes all types of DP-based techniques able to prevent membership inference attacks [63]. GDP can also bound the success of property inference, but it will lead to a considerable loss of the FL model utility if without a large number of local users [64], [65]. The GDP-based PPAgg approach in FL can add less noise than the LDP-based PPAgg method when guaranteeing the same privacy-preserving level. Still, it should satisfy the condition that the global server is trusted or a trusted third-party server is available. LDP-based PPAgg is more common and practical in FL, but this approach can not restrict the property inference attacks. There are also some new DP settings emerging with practical privacy loss accounting method, like (ϵ, δ)-DP [19], different versions of concentrated differential privacy (CDP [66], zCDP [67], tCDP [68]), and Rényi differential privacy (RDP) [69]. They are variants of the standard DP definition.

Note that OTP-based masking is different from DP-based perturbation. The reason is that OTP provides perfect secrecy, but DP still leaks some statistical information from the database. In addition, OTP-based, i.e., masking-based, aggregation usually provides exact results while DP-based aggregation suffers from noise, hence the degradation of FL model performance.

3.5 Blockchain

Blockchains was originated from the concept of cryptocurrencies, i.e., Bitcoin, to serve as a tamper-proof and decentralized ledger to record an ordered set of transactions in a transparent and immutable manner [20]. Specifically, these transactions are verified by trustless blockchain nodes through a decentralized consensus protocol and are constructed into blocks before attaching to the blockchain. Apart from the transactions, a block also contains a cryptographic hash of the previous block, which provides linkability and traceability. Here, we summarize the key advantages that blockchain networks can offer as follows:

- *Decentralization*: Each transaction to be attached to the blockchain must be confirmed upon the agreement among the majority of the blockchain nodes through a decentralized consensus protocol. As such, the single-point-monopoly of a centralized network can be removed from the blockchain.
- *Immutability*: The transactions stored in blockchain ledgers cannot be altered or tampered with unless the majority of nodes are compromised. Such security is guaranteed by the cryptographic techniques used in the blockchain that any change of the transaction data can be observed by all blockchain nodes.
- *Transparency* and *Auditability*: The transactions stored in blockchain ledgers are visible to all blockchain nodes and can be traced back for verification.
- *Pseudonymity*: By using the digital signature techniques, blockchain allows nodes to execute the transaction in an anonymous manner, without the intervention of any trusted third party.

To enable more complicated function evaluation, e.g., aggregation, over a blockchain rather than only data recording, smart contract [70] is utilized, which can be written into lines of code and automatically executed when predefined conditions are met. In general, the smart contract cannot be modified once it is deployed in the blockchain, and its execution is also decentralized, which ensures stable and reliable control functions [71]. Therefore, blockchain would be a natural choice for decentralized FL solutions. In specific, blockchain-based FL schemes will distribute the aggregation task from a single server to a set of blockchain nodes, which can mitigate risks of single-point failures and provide the auditability of the correctness of gradient collecting and model aggregation as well [72].

3.6 Trusted Execution Environment

The Trusted Execution Environment (TEE), as defined by GlobalPlatform, is a secure area of the main processor that allows the sensitive data and code to be stored, processed, and protected in an isolated and trusted environment [73]. In other words, TEE is isolated from the pure software environment, i.e., Rich Operating System Execution Environment (REE). Thus, TEE guarantees the confidentiality and integrity of insider applications and related data against any attacks from REE.

The implementations of TEE, e.g., [74], [75], are supported by hardware enclaves, such as Intel SGX [76] and ARM TrustZone [77], where a trade-off exists between the computation resource, e.g., limited memory size, and provided security level. Note that compared to REE-based applications, TEE usually involves extra costs regarding hardware which may hinder its large-scale deployment.

4 Privacy-Preserving Aggregation Protocols in Federated Learning

In this section, we will survey different constructions of PPAgg protocols, their applications in FL systems, and an extensive analysis of the advantages and disadvantages of these selected PPAgg protocols and solutions.

4.1 Masking-based Aggregation

Pair-wise masking. As mentioned in Section 3.1, OTP-based masking can be adopted to encrypt a message to preserve its privacy. For example, in a typical FL system, the users can mask their models and then upload them to the central server for aggregation. This requires well-designed protocols to enable the central server to obtain the aggregation results from these masked models, of which the research direction is arguably pioneered by the design of SecAgg [36]. In specific, assume there is a set of ordered FL users $\mathcal{U}$ where each $u_i \in \mathcal{U}$ has a vector $\boldsymbol{x}_i$. In SecAgg, each user u_i add a pair-wise additive mask to its vector $\boldsymbol{x}_i$ to get the masked vector $\boldsymbol{y}_i$.

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \sum_{u_j \in \mathcal{U}: i<j} \mathrm{PRG}(s_{i,j}) - \sum_{u_j \in \mathcal{U}: i>j} \mathrm{PRG}(s_{j,i})$$

where pseudorandom generator (PRG) can randomly generate a sequence numbers based on the seed $s_{i,j}$. It can be observed that the masks will be canceled when all masked vectors $\boldsymbol{y}_i$ are added such that

$$\sum_{u_i \in \mathcal{U}} \boldsymbol{y}_i = \sum_{u_i \in \mathcal{U}} \left(\boldsymbol{x}_i + \sum_{i<j} \mathrm{PRG}(s_{i,j}) - \sum_{i>j} \mathrm{PRG}(s_{j,i}) \right) = \sum_{u_i \in \mathcal{U}} \boldsymbol{x}_i$$

In addition, to handle the dropped users during the protocol execution, the Shamir secret sharing scheme (see Section 3.2) is used to share the seeds among users. Diffie-Hellman (DH) key exchange protocol [78] is adopted to make an agreement on the seed $s_{i,j}$ for each pair of user (u_i, u_j).

Note that the SecAgg scheme is not cost-effective for large-scale FL applications. For an n-user FL system, it requires $\mathcal{O}(n^2)$ communication-round to run the pair-wise DH key exchange protocol. Therefore, communication-reduction techniques to further reduce the overheads. For example, in [79], [80], well-designed quantization techniques, i.e., approximating high-bit numbers to low-bit numbers, are used to optimize the communication efficiency. In [81], the authors integrate the random rotation technique with SecAgg to aggressively adjust the quantization range of the users' models to reduce the model volume. CodedPaddedFL [82] adopts coding technique to improve the efficiency. In [83], heterogeneous quantization is introduced to adjust users' quantization level according to their available communication resources. In [84], gradient sparsification technique is adopted to compress the users' model. Besides, SecAgg-based PPAgg protocols for federated submodel learning can be found in [85] and [86]. However, the above-mentioned works rely on the SecAgg scheme for aggregation and thus still involve high communication overheads when it comes to large-scale FL systems. To reduce the communication overheads of SecAgg while keeping the use of the pair-wise masking technique, several follow-up schemes are proposed in which FL users communicate across only a subset of the user rather than all users. For example, TurboAgg [87] divides n FL users into $n/\log n$ groups and then follows a multi-group circular structure for aggregation. A similar grouping structure can be found in SwiftAgg [88]. However, these schemes require additional communication rounds to process between groups and sacrifice some security guarantees against colluding adversaries.

As such, aggregation schemes with a non-group architecture are considered. CCESA [89] allows FL users communicate, i.e., run the pair-wise DH key exchange protocol, over a sparse random graph, i.e., Erdös-Rényi graph, instead of the complete graph. An illustrative topology comparison between CCESA and SecAgg is given in Figure 2. Different from Erdös-Rényi graph, SecAgg+ [90] adopts Harary graph, a k-connected graph with graph vertices having the smallest possible number of edges. Both CCESA and SecAgg+ reduce the communication complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$.

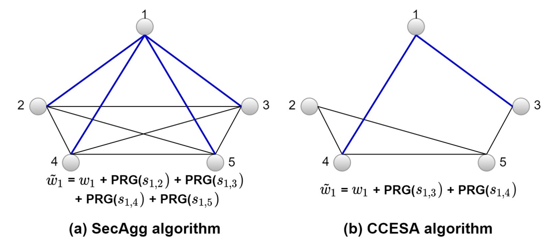


Fig. 2. The sparse communication graph of CCESA [89] compared to the complete communication graph of SecAgg [36]. We can observe that users in (b) communicate with much fewer users than those in (a).

Another direction to improve the efficiency of the pair-wise-masking-based aggregation scheme is to replace the DH key exchange protocol with a lightweight or non-interactive algorithm. For example, in Nike [91], with the assistance of two non-colluding cryptographic secret providers, i.e., aided servers, the seed of each pair of FL users can be generated in a non-interactive way. In [92], a trusted dealer is involved in assigning seeds to users of which the sum equals zero. In FLASHE [93], with the assumption that the semi-honest central does not collude with any single FL user, the authors propose a lightweight homomorphic encryption algorithm with a pair-wise masking style. However, the trust distribution of Nike [91] and FLASHE [93] are limited to the number of aided servers and non-colluding assumptions, hence limiting their application scenarios. Alternatively, a recent work [79] improves the efficiency upon SecAgg by removing the operations for secretly sharing seeds between FL users, resulting in a much lower communication cost if there is no dropped user, but a higher communication cost for the case of a large number of dropped users. Similar to [79], [94] removes seed secret sharing operations but requires all alive FL users to upload the shares of the whole masking vector.

Apart from protecting user privacy in a single FL round, several studies focus on the privacy issues caused by multiple-round FL training. For example, FedBuff [95] and LightSecAgg [96] allow asynchronous aggregation. In [97], the authors point out that even with the aforementioned privacy-preserving aggregation protocols, the multiple-round FL training may lead to severe information leakages due to the dynamic user participation. As shown in Figure 3, user u_1, u_2, u_3 participate in round t, and user u_1, u_2 participate in round $t+1$. If the model of u_1 and u_2 do not change significantly over the two rounds, the server can reconstruct the model of u_3 with a tiny error. This privacy issue is common in FL training when the global FL model converges. In [97], a naive mitigation method is proposed that requires the aggregation from a set of user batches rather than the individual users. Hence adversaries cannot differentiate the user models in the same batch for any long time.

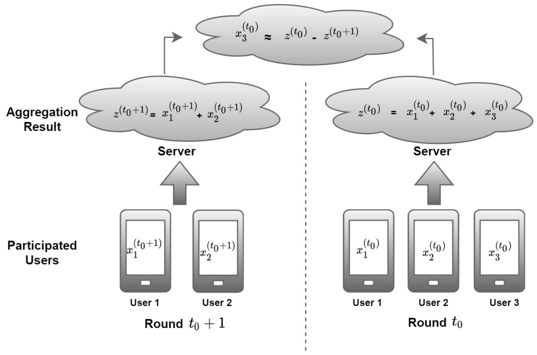


Fig. 3. A figure adapted from [97] to describe the information leakage due to the multi-round FL training.

Non-pair-wise masking. Although the pair-wise masking technique provides the attractive property that masks can be canceled when aggregating all the masked vectors, it naturally involves the interactions for the seed agreement between pairs of FL users, and thus hinders efficient deployments for large-scale FL systems. Therefore, some recent works replace pair-wise masking with lightweight non-pair-wise masking followed by one-shot unmasking. In particular, in such a scheme with n users, each FL user u_i generates its mask r_i without any interaction with others and uses it to encrypt its held vector x, then uploads the masked vector, e.g., $y_i = x_i + r_i$ in an additive masking manner, to the central server. Meanwhile, all users involve a protocol that allows the central server to know the sum of users' masks $R = \sum r_i$ while keeping the privacy of each r_i. In this case, the server is able to obtain the aggregation result by calculating $\sum x_i = \sum y_i - R$. For example, in HyFed [98], a trusted party is involved in calculating the aggregated noise from users, which then be sent to the server for one-shot unmasking. In [32], homomorphic PRG (HPRG) is adopted to achieve a lightweight mask generation, hence greatly reducing the communication overheads. Similar HPRG-based scheme is proposed in [99]. The idea of one-shot unmasking is also employed in [100] where a trusted third party coordinates dropped users and assists in calculating for unmasking. In the follow-up work LightSecAgg [96], the authors propose a lightweight and dropout-resilience secret sharing method, hence removing the trusted third party in [100] and allowing the server to do one-shot unmasking based on the received shares from alive FL users. Some other schemes achieve one-shot unmasking with a chain structure [101], [102], [103], [104]. In specific, as shown in Figure 4, assume a total order on all FL users u_i, a central server or leader generates a mask and assigns it to the first user u_1, which is used by u_1 to mask its model. After that, u_1 transfer its masked model to the user u_2, then u_2 adds its model to the masked model received from u_1 and transfer it to u_3. Following this way, all users aggregate their models one by one in a chain style, and the last user transfers its result to the central server for unmasking. However, these works assume that the central server is trusted and there is no collusion between the server and users, which is impractical for real-life FL applications.

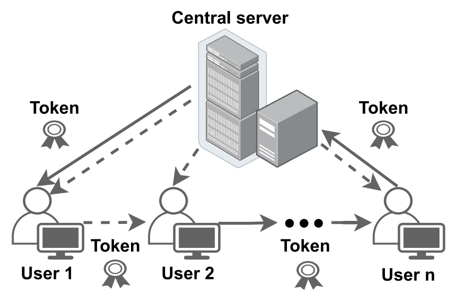


Fig. 4. Aggregation follows a chain structure where the token is transferred from user i to user $i + 1$, which is finally returned from user n to the central server.

Protecting the global model. Note that the above-mentioned masking-based aggregation protocols aim to protect the privacy of FL users' models or gradients. There are some other works that aim to protect the privacy of the global FL model. In [106], the server masks the global model and requires all FL users to do the training based on the masked model. After the training, each user sends some supplementary information to the server for unmasking. In PrivFL [107], a two-party computation technique is adopted, which allows each server-user pair to jointly train an ML model while preserving the privacy of both the global model and the user's local models. After that, each user masks its share and sends it to the server, and all users are involved in a privacy-preserving protocol, e.g., SecAgg [36] or SecAgg+ [90], to aggregate the sum of their masks, which then be used by the server for unmasking. Furthermore, some other techniques are considered for integration to improve the security during the whole aggregation. For example, in [108], differentially private noises are added to users' models to guarantee privacy during weighted averaging in aggregation. TEE in [79], pseudo-random functions in [109], MAC-like technique in [110], homomorphic hash in [112], zero-knowledge proofs in [113], and commitment scheme in [111] are deployed to guarantee that the server correctly aggregates the sum from FL users.

So far, we have reviewed masking-based aggregation protocols to protect users' model privacy and global model privacy. We should note that aggregation protocols based on pair-wise masking allow efficient unmasking with dropped users, and thus are suitable for the cross-device setting where FL users are mobile IoT devices that may drop out of the system at any time. In contrast, aggregation protocols based on one-shot unmasking usually improve efficiency for cross-silo settings. In most cases, there is a trade-off between security assumptions, e.g., a trusted party or non-colluding participants, and efficiency, e.g., computation, communication, or storage costs. Besides, to provide additional security apart from a privacy perspective, other cryptographic tools and trusted environments need to be considered. To summarize, we list the aforementioned aggregation schemes with related features in Table 2.

4.2 HE-based Aggregation

The HE-based aggregation in FL is relatively straightforward than that of masking-based aggregation. To aggregate

TABLE 2
A summary of masking-based PPAgg protocols.

Scheme	Masking type	Threat model	Privacy guarantee	Complexity discussion	Methods for security and efficiency improvements
[105] [36]	Pair-wise mask	Malicious users and server	Local model	Impractical for large-scale FL due to $\mathcal{O}(n^2)$ communication complexity	Baseline
[79]					Model quantization;TEE
[80] [81] [83]					Model quantization
[84]					Coding approach
[82]					Model spasification
[85] [86]				Determined by the size of submodels	Submodel aggregation
[87] [88]				Requires additional $\mathcal{O}(n/\ \log n)$ communication rounds	Group aggregation
[89] [90]				Reduce communication complexity to $\mathcal{O}(\log n)$	Sparse communication graph
[91] [92]				Replace user interactions with a trusted party	Trusted third party
[79]				Efficient with few dropped users	Removing secret sharing and introducing TEE
[94]		Semi-honest users and server		Efficient with low dropout rates	Replacing seed secret sharing with sending vectors
[95] [96]				Determined by the integrated PPAgg protocol	Asynchronous aggregation and lightweight reconstruction
[93]		Semi-honest server and users		Efficient in M1 setting	Lightweight HE
[97]		Semi-honest users and server	Multi-round privacy	Determined by the user selection strategy	Batch partitioning
[106]			Global model	Efficient without protecting local models	Mask global model
[107]			Local model; Global model	Determined by the 2PC protocol adopted	Two-party computation
[108]		Semi-honest users and server		Determined by the MPC protocol adopted	MPC and central DP
[109] [110] [111]		Semi-honest users and server	Local model	Involve additional complexity for verification based the selected techniques.	Methods of guarantee the aggregation correctness
[112]		Semi-honest users; Malicious server			
[113]		Malicious users; Semi-honest server			
[32] [99]	Non pair-wise mask	Malicious users and server	Local model	Efficient for small ML models	Homomorphic PRG
[98] [100]		Semi-honest users and server		Replace complicated unmasking with a trusted party	Trusted third party
[101] [102] [103] [104]				Additional n communication rounds for following the chain structure	Chain structure

the sum of users' locally trained models in an FL round, the users need to encrypt their models and send them to the central server. Then the central server adds the received encrypted models together, relying on the additive homomorphic property of the used crypto-system, which can be decrypted to obtain the global model in that FL round (see Figure 5).

The security of HE-based aggregation protocols in FL is achieved through their underlying crypto-system. According to the introduction of homomorphic encryption systems in Section 3.3, the homomorphic property is held only when the ciphertexts are encrypted using the same public key. Therefore, in a distributed setting such as FL, the key management of the crypto-system usually determines the threat model and application scenarios. More specifically, for a public-key cryptosystem used for aggregation in FL, the ownership of the secret key is the essential factor to be considered when it comes to industrial deployments. First, we summarize the different models of secret key management in FL.

- **M1**: Only known to all FL users and not to the central server.
- **M2**: Only known to the central server and not to any FL users.
- **M3**: Split across all or a subset of FL participants.

M1 setting. In the M1 setting, the secret key is known to all FL users but kept confidential against the central server. In this case, users' model privacy is protected, but the global model is public to all FL participants, and the M1 setting is only suitable that all FL users are honest and non-colluding. The secret key can be generated via interactions between users [114], [115], [116] or with the assistance of a trusted third party [117], [118], [119], [120], [121]. Besides, the crypto-system can be instantiated in different way to provide different security level, e.g., whether post-quantum or not, such as RSA-based [122], BGN-based [123], ElGamal-based [27], Paillier-based [114], [115], [118], [120], [124], [125], [126], [126], [127], lattice-based crypto-system [128], [129]. However, the above-mentioned schemes require rigorous security assumption that all users are at least semi-honest while there is no collusion between any user and the central server. This assumption is quite weak

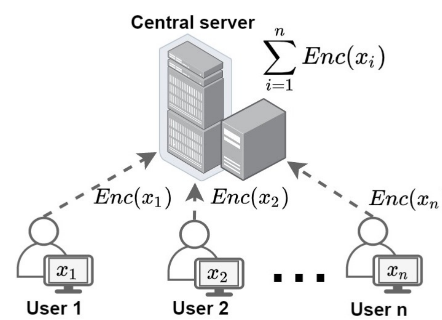


Fig. 5. An illustrative figure for the aggregation in FL based on homomorphic encryption.

since the server can directly break the security of users' model privacy by creating a pseudonymous FL user.

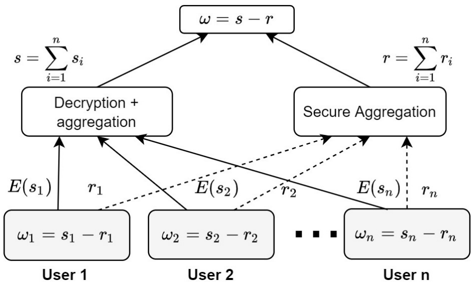


Fig. 6. An adaptive figure from PrivFL [107] for illustration.

M2 setting. Different from the M1 setting that provides users' model privacy, in the M2 setting, the secret key is held by the central server, which aims to protect the privacy of the global model from the server. Such consideration is common in many FL applications as the central server is usually a commercial company or consulting agency that intends to use FL to improve its ML model performance and provides model inference as a service for commercial profits [23], [33], [130]. In this case, the privacy of the global model is considered to be protected. However, as the central server holds the secret key, the encrypted model sent from FL users to the server can be decrypted by the server, which directly breaks the security of users' model privacy. Therefore, other privacy-preserving techniques are required to further improve the privacy guarantee. For example, in PrivFL [107] where only the server holds the secret key of the HE scheme, users mask their models before aggregation and involve in a SecAgg protocol to aggregate the masks for unmasking (see Figure 6), hence protects the privacy of both users' models and the global model. Other works such as [117], [122], [131], [132], [133], [134], [135] rely on a trusted party to manage the secret key. In PIVODL [136], an FL user is randomly selected as the key generator in each round, hence adding randomness to improve the security. However, the trust distributions of these works are still limited to only one party. Besides, since the global model is encrypted, FL users have to train their local model over ciphertext, hence involving FHE or MPC, which may lead to impractical computation or communication overhead, especially in the cross-device setting.

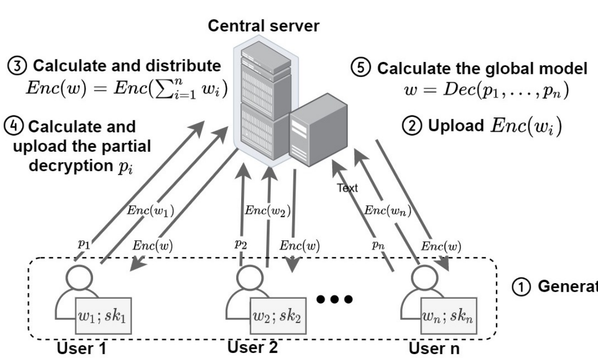


Fig. 7. An illustrative workflow of the aggregation in FL with a threshold addtive HE crypto-system.

M3 setting. Recall that in the M1 setting, all or a set of users hold the same secret key, and thus the ciphertext is no longer meaningful as long as the aggregation server colludes with any user that has the secret key. A straightforward method to deal with this security issue is to split the secret key across a set of users, i.e., the M3 setting. In this case, with a threshold cryptosystem (see Section 3.3), several users that more than a threshold number must cooperate in order to decrypt an encrypted message. Such a setting improves the security guarantee upon that of the M1 setting as the central server must collude with much more FL users to break the security but with high communication or computation costs. Therefore, the M3 setting is suitable for FL participants with sufficient communication bandwidth and computational power, i.e., a cross-silo setting. For example, as shown in Figure 7, in an FL system with a full-threshold crypto-system, all participants work as follows to achieve a privacy-preserving aggregation. (i) Users involve in a key generation protocol to generate a key-pair (p_k, s_k) that the public key p_k is known to everyone while the secret key s_k is shared. For example, for an n-user FL system, each user u_i has a partial secret key s_k^i such that $f(s_k^1, \ldots, s_k^n) = s_k$ where $f(\cdot)$ is determined according to the crypto-system used. (ii) Each user trains its local model w_i based on the global FL model, and encrypts it using the public key p_k and sends $Enc(w_i, p_k)$ to the central server. (iii) After receiving the encrypted models from users, the central server calculates $Enc(w, p_k) = \sum_{i=1}^{n} Enc(w_i, p_k) = Enc(\sum_{i=1}^{n} w_i, p_k)$ relying on homomorphic operations, and sends the result to users. (iv) Each user u_i uses its partial secret key to calculate a partial decryption $p_i = Dec_p(s_k^i, Enc(w, p_k))$ and sends it to the server, where Dec_p is determined according to the crypto-system used. (v) After receiving the partial decryptions from users, the server calculates $w = Dec(p_1, \ldots, p_n)$ to obtain the aggregation result w, i.e., the global model of the current FL round, where $Dec(\cdot)$ is determined according to the crypto-system used.

Partial HE schemes are widely used for the M3 setting. For example, threshold Paillier crypto-systems are adopted in [137], [138], [139], [140]. In [141], a more efficient variant of the threshold Paillier cryptosystem called BCP is introduced. Upon basic threshold Paillier crypto-systems, Helen [142] integrates zero-knowledge proof, and some supporting protocols of SPDZ [46] to guarantee the security against active malicious FL participants. An ElGamal-based crypto-system is deployed in [143] to improve the computation efficiency compared to the Paillier-based crypto-system. A lightweight AHE scheme for aggregation in FL is proposed in [144]. However, the above-mentioned partial HE schemes support only addition or multiplication. Therefore, they are suitable to be used only for the aggregation part, which involves only addition operations, instead of the functions that involve complex arithmetic operations.

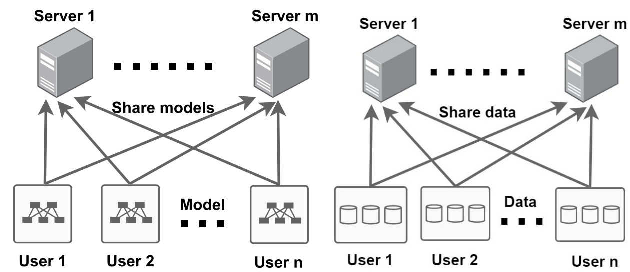
Protecting the global model. To further provide privacy guarantees of the global FL model rather than only the users' models during aggregation, the global FL models are required to be encrypted. In this case, users have to train their local models over ciphertext. Since the training process involves both addition and multiplication operations, fully homomorphic encryption crypto-systems that support arithmetic operations become necessary. Therefore, lattice-based FHE crypto-systems are adopted in FL schemes [30], [144],

[145], [146], [147] to enable more complicated function evaluation, e.g., ML model training, and extended to their multi-key versions for privacy purposes. For example, SAFELearn [30] describes a general PPFL scheme based on a multi-key FHE crypto-system. SPINDLE [146] proposes a PPFL scheme for the generalized linear ML model that protects the privacy of the whole FL workflow, i.e., the privacy of both users' models and the global model, relying on a multi-key version of CKKS crypto-system [58]. POSEIDON [147] extends the supported ML models of SPINDLE [146] from linear models to neural networks and comes up with a distributed bootstrapping protocol for training deep neural networks in an FL setting. Taking into account that FL systems with a standard multi-key lattice-based FHE crypto-system do not allow new FL users to join who do not participate in the public key generation, [145] involves a setup phase based on Shamir secret sharing where all users can exchange their shares of secret keys, hence new users are allowed to join by obtaining corresponding shares. To improve the efficiency, some other works [148], [149], [150] adopt multi-input functional encryption schemes. However, functional encryption involves high computation complexity for complicated functions, and a trusted party is needed for key generation and distribution, which weakens the security guarantee compared to the threshold and multi-key crypto-systems based FL.

We should note that the nature of threshold and multi-key crypto-systems inevitably leads to an expensive public key generation process that involves interactions between all users. Besides, protecting the privacy of the whole FL workflow requires the users to train their ML models over ciphertext, which is impractical for complicated functions such as deep neural networks, even with state-of-the-art techniques. Therefore, one must carefully weigh the trade-off between privacy guarantee and FL efficiency of such a setting according to the application scenario and its security requirements.

4.3 MPC-based Aggregation

Share model. For multi-party settings such as distributed machine learning and federated learning, MPC can be a natural option to enhance the security guarantee of the systems. A number of works employ MPC methods to achieve privacy-preserving aggregation and further privacy-preserving FL training. As shown in Figure 8(a), MPC-based privacy-preserving aggregation protocols allow FL users to distribute, i.e., share (see Section 3.2), their locally trained models to a set of agents, e.g., selected users or assistant servers. Then these agents jointly calculate the sum of users' models to obtain the share of the aggregation result. After that, they may choose to reconstruct the result, i.e., the new global FL model. For example, in [151], [152], each FL user shares its locally trained model with all users for aggregation using generic MPC protocols. In Fastsecagg [153], by sacrificing some security, the authors substitute the standard Shamir secret sharing with a more efficient FFT-based multi-secret sharing scheme. Alternatively, models can be shared between two servers as in [154], [155], [156], [157], or several servers such as [158], [159]. Some other works introduce a two-phase secret-sharing-based aggregation [160], [161]. In the first phase, all users are involved in an MPC-enabled selection protocol to construct a committee. Then in the second phase, all users share their models with the users in the committee for aggregation, relying on standard MPC protocols similar to the works mentioned above. Furthermore, to guarantee the correctness of the aggregation result with a malicious central server, verifiable secret sharing schemes are adopted in [159], [162].



(a) Aggregation (b) Training

Fig. 8. MPC-based privacy-preserving aggregation and training where users' models or data can be shared among FL participants.

Share data. Similarly, if FL users distribute their local data to several servers for training instead of aggregation, as shown in Figure 8(b), the scheme boils down to the MPC-based distributed ML training. Note that such a scheme directly supports privacy-preserving aggregation as the locally trained models are also shared among MPC participants, i.e., servers or selected users. In general, such privacy-preserving training can be achieved by splitting the trust among two [45], [48], three [47], [49], four [50], [51], or any number of servers, respectively, relying on generic MPC protocols. For example, [163] adopts ABY [45] and SPDZ [46] to achieve a secure federated transfer learning. [164] and [165] rely on a 2PC protocol. [166] deploys a gradient tree boosting model based on GMW protocol. Multi-party settings are considered in [142], [167] where SPDZ protocols are applied. Note that these works are the applications of generic MPC protocols to FL, of which the efficiency and security improvements mainly come from their underlying MPC schemes. We refer interested readers to [168] for the details of state-of-the-art MPC protocols.

However, the nature of MPC limits its application to FL. Firstly, sharing secrets among all users usually leads to a significant communication overhead. Thus, for a practical pure-MPC-based FL system, the data or models of FL users have to be transferred to a small number of MPC participants. In this case, security can be guaranteed only when the MPC participants do not collude, or the majority of them are trusted. Furthermore, the above-mentioned MPC-based privacy-preserving training schemes usually require all MPC participants to keep online during the execution of the whole MPC protocol, and possess sufficient bandwidth and computational power. Therefore, these MPC participants cannot be resource-constrained mobile or IoT devices that may drop out of the system at any time, which is common in a cross-device FL setting.

Handle dropped users. Apart from the above-mentioned works that models or data are directly shared among MPC participants to protect privacy, secret sharing

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2022.3190835

schemes can also be adopted as supporting protocols to handle dropped users or to verify computation results in FL systems. As described in Section 3.2, a threshold secret sharing scheme, e.g., Shamir secret sharing, can be a natural choice to enable protocol execution even with a set of users of which the number is greater than a threshold value. If these users are set to be online users, the proposed protocol will obtain a dropout resilience. For example, in [32], [36], [89], [90], the seeds for mask generation are shared using the Shamir scheme, which allows the central server to reconstruct the masks of dropped users if the number of online users is greater than the threshold. If these users are set to be honest users, the proposed protocol will allow the verification of the computation results. For example, [159], [162] acknowledge the verification of the correctness of the aggregation result only when a set of users of which the number is greater than the threshold have verified the result. However, we should note that the setting of threshold value leads to a trade-off depending on its usage. A larger threshold means a more robust security guarantee but less protocol efficiency, and vice versa.

4.4 DP-based Aggregation

Unlike the cryptographic approaches, DP-based perturbation methods require less computational overhead to achieve PPAgg in FL. This subsection will review related research works that leverage DP-based techniques to guarantee information privacy in FL.

LDP. LDP-based PPAgg protocols are typically adopted in the setting with an untrusted server. To protect data privacy, users need to perturb their locally trained models or gradients before sending them to the central server. For example, [169] leverages the Gaussian mechanism to guarantee ϵ differentially privacy for the local models in an Internet of Vehicles (IoV) scene. With a similar setting, [170] proposes a novel LDP model and integrates it with the FedSGD algorithm [171] to construct an LDP-FedSGD scheme, bounding the success of membership inference. SFSL [172] applies LDP on submodel update to hide users' index-related information in an FL system for e-commerce recommendation. Following a client-edge-cloud structure, [173] hierarchically applies Gaussian noise to the three kinds of FL participants. Laplace noise is considered to be adopted to perturb the non-discrete parameters in users' local models [174], [175]. Note that LDP-based PPAgg protocols require adding noise to the target parameters in each FL round. Thus, the total number of users, the number of users selected in each FL round, and the number of iterations will affect the methods and amount of DP noise allocation, hence the model usability. The research works [176], [177], [178], [179] propose methods taking these factors into account to further improve the model usability in practical FL systems.

GDP. In general, LDP-based methods usually involve more noise, hence more severe model performance degradation, compared to GDP-based methods. Therefore, when a trusted server is considered, GDP can be integrated to provide user-level privacy and a better model utility by perturbing the global model on the server-side. The first work that considers GDP in an FL setting is described in [180], which aims to protect the whole dataset's privacy, and Gaussian noise is adopted. Similarly, [181] investigates the effect of GDP on the performance of FedAvg and FedSGD algorithms. Both [180], [181] show that the privacy of the trained FL model can achieve better accuracy with a large number of FL users. Upon using the perturbing technique based on Gaussian noise, [182] proposes a K-client random scheduling strategy to select users for FL model training for better model usability. Noisy-FL [183] leverages a privacy tracking framework f-DP [62] to track the privacy loss and hence adjusts the GDP-based PPAgg protocol to lower the requirements on a large number of clients. Furthermore, considering that dropped users can rejoin the FL training in a heterogeneous IoT setting, the authors in [184] design a personalized PPFL model to achieve (ϵ, δ)-DP with L_2-sensitivity by adding the Gaussian noise to the global model during each iteration.

Hybrid methods. To protect the privacy of both users' locally trained models and the global model, a combination of LDP and GDP for a hybrid PPAgg construction becomes a natural option. For example, [185] adopts both LDP and GDP in a joint manner. Furthermore, one can consider the DP-based technique with other PPTs to provide additional security guarantees. The authors in [186] apply additive HE and DP with the Gaussian mechanism to prevent privacy leakage from the local gradients and the shared models in FL, respectively. Their method can protect data privacy even when the attacker colludes with multiple participants. RDP and MPC are utilized in [187] where the data distributions are similar among different users, and the designed model is suitable for non-i.i.d (independent and identically distributed) data. The combination of MPC and DP can also be found in [188]. LDP and function encryption are combined in [189] to achieve both data-level and content-level privacy-preserving. Compression and SecAgg are combined with DP in [190] to guarantee both private and accurate models by an adaptive quantile clipping method. LDP and shuffled models are utilized in [191], [192], [193] to enhance model security by amplifying privacy through anonymization. An asynchronous model update scheme and a malicious node detection mechanism are designed to integrate with LDP in [194] for communication-efficient and attack-resistant Federated Edge Learning. In [195], the Skellam mechanism instead of the Gaussian mechanism is introduced, and the authors explore its performance when combining it with central RDP, distributed RDP with secure encryption, respectively. The authors in [196] combine LDP with secure encryption and zCDP to achieve a good utility-privacy trade-off by adding less noise in every training iteration.

In a nutshell, DP-based PPAgg protocols inevitably result in a trade-off between data privacy and model usability, including but not limited to convergence performance, communication efficiency, and accuracy. Thus, one needs to determine appropriate methods for privacy budget distribution and consider properly combining other PPTs to improve model usability in different FL settings.

4.5 Blockchain-based Aggregation

The nature of blockchain allows one to distribute the trust from a single server to a set of blockchain nodes, hence providing resilience to single-point-of-failure. In this case, a task publisher is usually involved in initializing the global FL model. Meanwhile, blockchain guarantees the auditability of the data and operation processed in the blockchain and the anonymity of participants (see Section 3.5). As shown in Figure 9, a typical blockchain-based aggregation protocol in FL works as follows: (i) the task publisher publishes the FL training task with an initial global model to the blockchain, then (ii) each FL user fetches the global model, and (iii) trains its local model. After that, (iv) each FL user generates and broadcasts a transaction recording its local model, which will be received and stored by nodes in their transaction pools. Then (v) the elected consensus node aggregates those local models to obtain the global model via the consensus protocol. Finally, (vi) the new global model is included in a block attached to the blockchain for the next round of FL training.

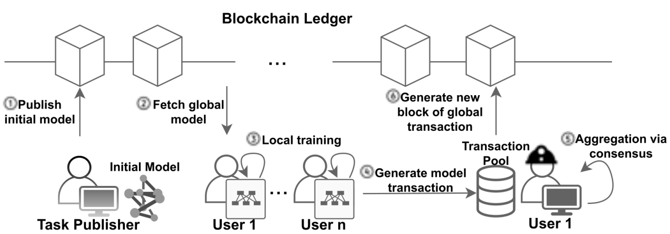


Fig. 9. An illustrative figure of the aggregation in FL based on blockchain.

Integration of PPTs. Note that the standard blockchain cannot prevent information leakage during model aggregation in FL, as users' models uploaded to the blockchain are still over plaintext. Thus, additional PPTs are considered to be integrated to achieve the blockchain-based PPAgg protocol. For example, CDP methods are adopted in [31], [197], [198], [199] and LDP methods are adopted in [200], [201], [202] to perturb users' models before uploading. Paillier crypto-systems are applied to [203], [204], [205] with its standard version as in the M1 setting, or with the threshold version [72] in the M3 setting (see Section 4.2). However, security concerns still exist regarding the key management, i.e., arrangement of the ownership of secret keys used in adopted crypto-systems. Therefore, several studies aim to introduce a third party as an assistant. For example, in [206], the task publisher and the third party have the key-pair (pk_1, sk_1) and (pk_2, sk_2), respectively. They cooperate to generate a public key pk_3 that messages encrypted by pk_3 can be transformed, i.e., re-encrypted, to the one encrypted by pk_1 in some way using sk_2. In this case, FL users encrypt their models using the public key pk_3 and send them to the third party via blockchain, where re-encryption and aggregation are performed. Thus, security can be guaranteed if the task publisher and third party are non-colluding. Apart from the above-mentioned works where users' models are uploaded or recorded by blockchain, several studies aim to leverage blockchain to store intermediate materials of existing PPAgg protocols for auditability. For example, [207] and [208] rely on the SecAgg protocol [36] where secret keys involved are shared and stored using blockchain.

Smart contract. Beyond integrating PPTs to provide privacy guarantees upon blockchain, the smart contract can be adopted to further enhance the security or efficiency of aggregation. For example, HE-based PPAgg protocol [209], [71] and [210] additionally leverage smart contract to generate the key-pair (pk, sk) where the public key pk is used to encrypt users' models while the secret key sk is passed to a trusted party [211], a leader elected by the blockchain consensus protocol [209], or a key manager in smart contract [71], [210]. These schemes regarding key management are similar to those in the M1 setting discussed in Section 4.2. However, the security risk is distributed to blockchain nodes, and the collusion risk is mitigated by using the smart contract.

In general, blockchain can be a practical choice for deploying decentralized aggregation, i.e., distributing the aggregation task to a set of blockchain nodes, and providing additional security guarantees due to its nature [201], [212]. Furthermore, originating from the concept of cryptocurrency, blockchain-based PPAgg makes it straightforward to integrate incentive mechanisms to motivate more data owners to contribute their data and mitigate some privacy issues from a game-theoretical perspective [213].

4.6 TEE-based Aggregation

As shown in Figure 10, a typical TEE-based aggregation protocol in FL works as follows: (i) all users encrypt their locally trained models and send them to REE, (ii) TEE loads received encrypted models from REE, then (iii) decrypts and aggregates them. After that, (iv) TEE outputs the aggregation result to REE for the distribution to all users. The adoption of such typical TEE-based PPAgg can be found in [214], [215], [216]. To further enhance the security against potential attacks to TEE [217], DP techniques are adopted to perturb users' models before uploading them to TEE [215]. Alternatively, in [218], the trust is distributed among several TEEs. In MixNN [219], a TEE is involved in shuffling users' models before uploading them to the central server, hence introducing randomness to enhance the security.

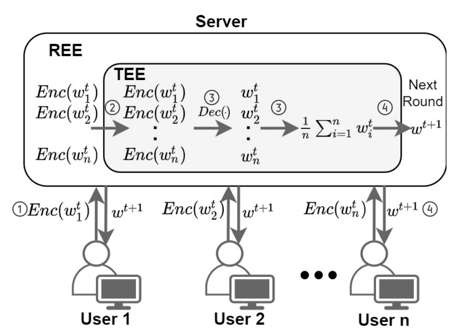


Fig. 10. An illustrative figure of the aggregation in FL based on TEE.

To extend TEE-based aggregation to be used in the training process to further protect the privacy of the whole FL workflow, several studies aim to deploy ML training algorithms in TEE environments [2], [215], [220], [221], [222]. However, the constrained memory size of the TEE platform usually leads to partial training in TEE [215], [220], [221], or fully training in TEE with a long-time delay [222].

Besides, signature and verification schemes can be integrated with TEE to provide integrity guarantees of honestly local training [214], [223], [224], or correct aggregation of the central server [79], [224]. However, we should note that as TEE-based PPAgg usually involves extra costs regarding

TABLE 3
A summary of PPAgg constructions. The provided security level, i.e., threat model, of the listed constructions is determined by their underlying protocols. Note that the “large” and “small” to describe the distribution scale correspond to those for cross-device and cross-silo FL settings.

	Masking	HE			MPC		DP		Blockchain	TEE
		AHE	Threshold HE	FHE	Additive	Shamir	GDP	LDP		
Privacy guarantee	Local model	Local model	Local model	Local model; Global model	Local model; Global model	Local model	Global model	Local model	N.A.	Local model; Global model
Distribution scale	Large	Large	Small	Small	Small	Large	Large	Small	Large	Small
Resource requirement	Usually involves interactions among users	Lightweight	Requires expensive key generation and decryption	Impractical computation overheads for large-scale ML models	Usually involves large communication overheads	Lightweight	Negligible		Costs of transactions	Requires the deployment of TEE hardware platforms
Dropout-resilience	Support	Support	Partially support	Support	NO	Support	Support		Support	Support
Model utility	Depends on the encoding methods of underlying related cryptographic blocks						Affected		N.A.	N.A.
Example application	[32], [36], [90]	[114], [115]	[138], [142]	[146], [147]	[163], [167]	[36], [90]	[181], [182]	[170], [191]	[72], [198]	[177], [214]
Remark	Needs specific desgin for different FL settings	Key managements affect security		Impractical for deep neural networks	Suitable for only cross-silo FL settings	Widely used as a building block to handle dropped users	Suffers the loss of FL model performance		Needs to integrate with other PPTs	Limited computation resource and costs of hardware

hardware compared to the pure software-based PPAgg, large-scale deployment of TEEs in FL would be costly.

4.7 Discussions

So far, we have reviewed PPAgg protocols with the discussions on the advantages and disadvantages of their constructions. Since FL systems are deployed under different settings with respect to the threat model, resource requirement, distribution scale, etc., we summarize the relationships between and properties of these PPAgg constructions in Table 3 as a reference for interested readers that aim to design their customized PPFL schemes.

To provide privacy guarantees in an FL system, DP-based PPAgg is the most lightweight option but suffers from the loss of model performance. Thus, to avoid the performance loss, one has to adopt more expensive cryptographic tools, e.g., HE or MPC, to construct PPAgg to obtain the exact solution. Additive HE is the most natural option among those cryptographic tools as it directly supports aggregation. However, privacy issues exist due to the key management. In a standard HE scheme, neither the central server nor users keeping the secret key provide security against collusion. Threshold HE schemes with more robust key management inevitably involve the expensive key generation and decryption protocols. Besides, enabling the privacy protection of the whole FL workflow requires the adoption of FHE, which may lead to impractical overheads for large-scale ML models. Compared to HE, MPC is more efficient but still suffers from large communication overheads. Therefore, PPAgg protocols based on pure HE or MPC can be considered only in a cross-silo FL setting where participants have sufficient computation and communication capability and keep online from round to round. Compared to PPAgg based on pure cryptographic tools, masking-based PPAgg protocols are a more promising construction for large-scale PPFL schemes. They combine several lightweight cryptographic techniques, hence providing cost-effective execution for resource-constrained FL users. Additionally, Shamir secret sharing schemes are usually adopted to handle dropped users. However, the complicated constructions and lack of generality of masking-based PPAgg may hinder their wide deployment. To further improve the privacy and security of FL schemes, one can consider hybrid methods that integrate PPTs, blockchain, or TEE to provide desired properties.

In general, for the setting with resource-constrained FL participants, DP-based and masking-based methods are the most cost-effective options, where the former is more suitable for FL systems with tolerance with model accuracy loss while the latter requires cryptographic experts to customize the protocol design. Generic solutions of PPAgg without the degradation of model performance are cryptographic frameworks based on HE or MPC. One can choose one of them or make a combination to construct a PPAgg protocol according to the system's bottleneck, i.e., communication bandwidth, computational power or both. Blockchain and TEE are considered to provide additional security or privacy guarantee beyond the target of FL.

5 Federated Learning Frameworks for Privacy-Preserving Aggregation

With the development of federated learning, many FL frameworks have been proposed as open-source libraries to support follow-up works and enable easy deployment and replicability of FL systems. In Table 4, we list some existing open-source FL frameworks that support privacy-preserving aggregation with a summary of the construction of PPAgg protocols, privacy guarantees, and threat models. In particular, FATE [226] initiated by Webank's AI department based on Tensorflow is the most active developing framework in the FL community and supports the various privacy-preserving techniques, including MPC, encryption and adapted SecAgg protocol. TFF [225] supports only a DP-based method for privacy-preserving aggregation. PaddleFL [229] is developed based on PaddlePaddle and its cryptographic privacy preservation is achieved leveraging some existing MPC frameworks such as ABY3 [47] and PrivC [230]. In addition to integrating generic MPC frameworks, PySyft [231] is the only one that supports FHE. Flower [232] deploys SecAgg and SecAgg+ for privacy-preserving aggregation. Due to the different construction, one needs to select a proper FL framework to deploy Horizontal FL (HFL) or Vertical FL (VFL). Besides, only some of them are supportive of cross-device settings.

TABLE 4
Some open-source FL frameworks support PPAgg. A PPFL framework may consist of several PPAgg protocols for different settings, e.g., Horizontal FL (HFL) and Vertical FL (VFL), which provide different privacy guarantees. In general, the number of participants in VFL is less than that of HFL. Note that all listed frameworks provide a privacy guarantee on users' locally trained modes, and some of them support privacy-preserving training, hence protecting the privacy of the global model. If the PPAgg constructions provide security against active malicious settings, e.g., SPDZ and SecAgg, the framework also allows corresponding extensions.

Framework	PPAgg construction	Privacy guarantee	Threat model	FL setting	Remark
TFF [225]	Central DP [61]	Global model	Semi-honest users	Cross-device	Partial protection of user model privacy; Available for only large-scale FL.
FATE [226]	Paillier [52]; SPDZ [46]; OT [227]; VSS [228]; SecAgg [36]	Local model; Global model	Semi-honest users and server	Cross-silo	Paillier and SecAgg for PPAgg protocols in HFL; SPDZ and OT for PPAgg protocols in VFL.
PaddleFL [229]	Central DP [61]; SecAgg [36]; ABY3 [47]; PrivC [230]	Local model; Global model	Semi-honest users and server	Cross-silo	Central DP and SecAgg for PPAgg protocols in HFL; Generic MPC protocols, i.e., ABY3 for MPC and PrivC for 2PC, for PPAgg protocols in VFL.
PySyft [231]	Central DP [61]; SPDZ [46]; CKKS [54]; Paillier [52]	Local model; Global model	Semi-honest users and server	Cross-silo; Cross-device	Integration with PyGrid API for the FL mode; Supporting specific deployments on Android and iOS.
Flower [232]	SecAgg [36]; SecAgg+ [90]	Local model	Semi-honest users and server	Cross-silo; Cross-device	Mainly designed for large-scale FL settings with heterogeneous participants.

Note that in addition to the PPFL frameworks listed in Table 4, there are also PPFL frameworks under development from some leading IT companies and organizations, e.g., FedML [233], PrivacyFL [234], HyFed [98], Federated Learning and Differential Privacy framework by Sherpa.ai [235], Hive by Ping An Technology [236], Fedlearn-Algo by JD Finance [237], Huawei Noah's Ark FL framework [238], and some distributed with proprietary or limited licenses, e.g, the FL framework by Ant Group [166], NVIDIA Clara [239] and IBM-FL [240].

6 Challenges and Future Directions

In Section 4, we provide an in-depth survey on applications of PPAgg protocols to address a wide range of privacy and security issues in FL systems. However, with the fast evolution of PPFL schemes and their deployments, a plethora of emerging problems remain open for further studies, many of which require new PPAgg protocols to provide additional properties and support more operations. In this section, we expand our discussion to some challenges as well as research directions with PPFL systems, where PPAgg protocols may exert their further potential.

6.1 Throughput Improvement

The privacy-preserving aggregation protocols have been adopted in a lot of PPFL schemes. However, the throughput, i.e., the capacity of processing aggregation operations, of many PPAgg protocols limits the scope of PPFL applications, especially for large-scale networks. The reason is that their building blocks leverage generic expensive cryptographic techniques, which involve large computation or communication overheads. Thus, specific lightweight cryptographic techniques designed for aggregation in FL are required, e.g., communication-efficient masking-based algorithm [90], and lightweight additive HE (AHE) scheme [93]. Besides, as machine learning algorithms typically consist of a large number of vector operations, efficient integration of batch operations should be considered to improve the throughput, e.g., batch encryption [124], SIMD techniques in HE schemes [54], and parallelized hardware architectures such as FPGA and GPU. Furthermore, the combination of compression techniques with efficient PPAgg protocols can significantly reduce the communication overheads in FL but remains a challenge. This is because compression techniques, e.g., Top-k sparsification [241], require the order information to reconstruct the original vector from the compressed vector. Such information vector may lead to severe privacy leakage, while reconstruction over encrypted order information vector is not straightforward. Therefore, proper integration of different techniques with PPAgg protocols for throughput improvement can be further investigated.

6.2 Hybrid Schemes for Stronger Security

In addition to the privacy threats discussed in this survey, other attacks from a security perspective may also hinder the deployment of FL systems, e.g., poisoning attacks [242] and inference attacks [63]. Thus, integrating PPAgg protocols with other security algorithms to construct hybrid schemes can be considered for future research.

Poisoning attack. Poisoning attacks aim to reduce the accuracy of the FL model, i.e., random attacks [242], or induce the FL model to output the target label specified by the adversaries, i.e., targeted attacks or backdoor attacks [37], by manipulating local data or models. Therefore, PPAgg protocols cannot provide security against poisoning attacks. For poisoning attacks from the user-side, the central server can adopt Byzantine-resilient aggregation algorithms, e.g., [38], [39], [40] to detect anomalies. However, these schemes require access to the users' data or models, which violates the privacy goal of PPFL, and hence cannot be integrated with PPAgg protocols. Therefore, only Byzantine-resilient aggregation algorithms that do not access users' data or models such as [243] or those that work over encrypted data can be adopted for integration, which remains for further investigation. For poisoning attacks from the server-side, one needs to guarantee that the server correctly aggregates the models from users. TEE, blockchain, verifiable secret sharing (VSS), and verifiable computation techniques can be applied to PPAgg protocols for verification.

Inference attack. Inference attacks aim to cause information leakages of users' data, e.g., membership [63], attribute [64], or data [16], by probing a target ML model. Most PPAgg protocols mitigate this issue by protecting the users' models. However, privacy leakages still exist in some PPFL systems with PPAgg protocols. For example, as pointed out in [65], LDP-based aggregation does not guarantee security against attribute inference attacks, while GDP-based aggregation works only when sacrificing significant utility. Besides, in many PPFL systems, global models are revealed to adversaries, which are still vulnerable to inference attacks. Therefore, integrating other privacy-preserving techniques with PPAgg protocols to enhance security remains a topic for further research.

7 Conclusions

This paper has presented a comprehensive survey of the privacy-preserving aggregation protocols adopted to enhance the privacy of federated learning systems. Firstly, we gave an overview of federated learning on its concepts, data organization, working mechanism, and privacy threats to FL systems. Then, we introduced the basic knowledge of supporting tools for constructing PPAgg protocols. Afterward, we provided reviews and analyses of different constructions of PPAgg protocols in detail to deal with various privacy issues in FL systems. Finally, we outlined existing challenges and several directions for future research.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[2] A. Huang, Y. Liu, T. Chen, Y. Zhou, Q. Sun, H. Chai, and Q. Yang, "Starfl: Hybrid federated learning architecture for smart urban computing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 4, pp. 1–23, 2021.

[3] H. Yang, K.-Y. Lam, L. Xiao, Z. Xiong, H. Hu, D. Niyato, and V. Poor, "Lead federated neuromorphic learning for edge artificial intelligence," 2022.

[4] X. Li, L. Cheng, C. Sun, K.-Y. Lam, X. Wang, and F. Li, "Federated-learning-empowered collaborative data sharing for vehicular edge networks," *IEEE Network*, vol. 35, no. 3, pp. 116–124, 2021.

[5] Y. Liu, J. James, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet of Things Journal*, 2020.

[6] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, 2020.

[7] Z. Liu, Y. Chen, Y. Zhao, H. Yu, Y. Liu, R. Bao, J. Jiang, Z. Nie, Q. Xu, and Q. Yang, "Contribution-aware federated learning for smart healthcare," in *Proceedings of the 34th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-22)*, 2022.

[8] C. Ju, D. Gao, R. Mane, B. Tan, Y. Liu, and C. Guan, "Federated transfer learning for eeg signal classification," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 3040–3045.

[9] S. Chen, D. Xue, G. Chuai, Q. Yang, and Q. Liu, "Fl-qsar: a federated learning-based qsar prototype for collaborative drug discovery," *Bioinformatics*, vol. 36, no. 22-23, pp. 5492–5498, 2021.

[10] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, and Z. Shen, "Dealing with label quality disparity in federated learning," in *Federated Learning*. Springer, 2020, pp. 108–121.

[11] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "Fedvision: An online visual object detection platform powered by federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[12] ——, "Federated learning-powered visual object detection for safety monitoring," *AI Magazine*, vol. 42, no. 2, pp. 19–27, 2021.

[13] J. Luo, X. Wu, Y. Luo, A. Huang, Y. Huang, Y. Liu, and Q. Yang, "Real-world image datasets for federated learning," *arXiv preprint arXiv:1910.11089*, 2019.

[14] B. Tan, B. Liu, V. Zheng, and Q. Yang, "A federated recommender system for online services," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 579–581.

[15] L. Yang, B. Tan, V. W. Zheng, K. Chen, and Q. Yang, "Federated recommendation systems," in *Federated Learning*. Springer, 2020.

[16] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

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

[18] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 1982, pp. 160–164.

[19] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, 2014.

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[21] GlobalPlatform, "TEE system architecture," Available: http://www.globalplatform.org/specificationsdevice.asp, 2011.

[22] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, 2021.

[23] Q. Yang, "Toward responsible ai: An overview of federated learning for user-centered privacy-preserving computing," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2021.

[24] L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *arXiv preprint arXiv:2012.06337*, 2020.

[25] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[26] C. Briggs, Z. Fan, and P. Andras, "A review of privacy-preserving federated learning for the internet-of-things," *Federated Learning Systems*, pp. 21–50, 2021.

[27] C. Fang, Y. Guo, Y. Hu, B. Ma, L. Feng, and A. Yin, "Privacy-preserving and communication-efficient federated learning in internet of things," *Computers & Security*, vol. 103, p. 102199, 2021.

[28] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.

[29] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[30] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame *et al.*, "Safelearn: Secure aggregation for private federated learning," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 56–62.

[31] A. Mondal, H. Virk, and D. Gupta, "Beas: Blockchain enabled asynchronous & secure federated machine learning," *arXiv preprint arXiv:2202.02817*, 2022.

[32] Z. Liu, J. Guo, K.-Y. Lam, and J. Zhao, "Efficient dropout-resilient aggregation for privacy-preserving machine learning," IEEE Transactions on Information Forensics and Security, 2022.

[33] Y. Jin, X. Wei, Y. Liu, and Q. Yang, "Towards utilizing unlabeled data in federated learning: A survey and prospective," *arXiv preprint arXiv:2002.11545*, 2020.

[34] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," IEEE Transactions on Neural Networks and Learning Systems, 2022.

[35] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li, "The right to be forgotten in federated learning: An efficient realization with rapid retraining," *arXiv preprint arXiv:2203.07320*, 2022.

[36] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[37] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[38] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[39] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018.

[40] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," in *28th Annual Network and Distributed System Security Symposium, NDSS*, 2021.

[41] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2020.

[42] A. C.-C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. IEEE, 1986, pp. 162–167.

[43] M. O. Rabin, "How to exchange secrets with oblivious transfer," *Cryptology ePrint Archive*, 2005.

[44] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game, or a completeness theorem for protocols with honest majority," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, 2019, pp. 307–328.

[45] D. Demmler, T. Schneider, and M. Zohner, "Aby-a framework for efficient mixed-protocol secure two-party computation." in *NDSS*, 2015.

[46] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Annual Cryptology Conference*. Springer, 2012, pp. 643–662.

[47] P. Mohassel and P. Rindal, "Aby3: A mixed protocol framework for machine learning," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018.

[48] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 19–38.

[49] S. Wagh, D. Gupta, and N. Chandran, "Securenn: 3-party secure computation for neural network training." *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 3, pp. 26–49, 2019.

[50] M. Byali, H. Chaudhari, A. Patra, and A. Suresh, "Flash: fast and robust framework for privacy-preserving machine learning," *Proceedings on Privacy Enhancing Technologies*, 2020.

[51] H. Chaudhari, R. Rachuri, and A. Suresh, "Trident: Efficient 4pc framework for privacy preserving machine learning," *arXiv preprint arXiv:1912.02631*, 2019.

[52] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999.

[53] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[54] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 409–437.

[55] H. Tian, C. Zeng, Z. Ren, D. Chai, J. Zhang, K. Chen, and Q. Yang, "Sphinx: Enabling privacy-preserving online learning over the cloud," in *IEEE Symposium on Security and Privacy (SP)*, 2022.

[56] D. Rotaru, N. P. Smart, T. Tanguy, F. Vercauteren, and T. Wood, "Actively secure setup for spdz," *Journal of Cryptology*, 2022.

[57] T. Nishide and K. Sakurai, "Distributed paillier cryptosystem without trusted dealer," in *International Workshop on Information Security Applications*. Springer, 2010, pp. 44–60.

[58] C. Mouchet, J. R. Troncoso-Pastoriza, and J.-P. Hubaux, "Multiparty homomorphic encryption: From theory to practice." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 304, 2020.

[59] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2006, pp. 486–503.

[60] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[61] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[62] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *arXiv preprint arXiv:1905.02383*, 2019.

[63] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[64] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[65] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," *arXiv preprint arXiv:2009.03561*, 2020.

[66] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *arXiv preprint arXiv:1603.01887*, 2016.

[67] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography Conference*. Springer, 2016, pp. 635–658.

[68] M. Bun, C. Dwork, G. N. Rothblum, and T. Steinke, "Composable and versatile privacy via truncated cdp," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018.

[69] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017.

[70] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, 2014.

[71] X. Wu, Z. Wang, J. Zhao, Y. Zhang, and Y. Wu, "Fedbc: blockchain-based decentralized federated learning," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. IEEE, 2020, pp. 217–221.

[72] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2438–2455, 2019.

[73] GlobalPlatform, "Introduction to trusted execution environments," *Global Platform*, 2018.

[74] K. Kostiainen *et al.*, "On-board credentials: An open credential platform for mobile devices," 2012.

[75] Linaro, "OP-TEE," https://optee.readthedocs.io/en/latest/, accessed: 2022-03-23.

[76] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution." *Hasp@ isca*, 2013.

[77] L. ARM, "Arm security technology-building a secure system using trustzone technology," PRD-GENC-C. ARM Ltd. Apr.(cit. on p.), Tech. Rep., 2009.

[78] W. Diffie and M. E. Hellman, "New directions in cryptography," in *Secure communications and asymmetric cryptosystems*. Routledge, 2019, pp. 143–180.

[79] Y. Zheng, S. Lai, Y. Liu, X. Yuan, X. Yi, and C. Wang, "Aggregation service for federated learning: An efficient, secure, and more resilient realization," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[80] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, 2021.

[81] K. Bonawitz, F. Salehi, J. Konečnỳ, B. McMahan, and M. Gruteser, "Federated learning with autotuned communication-efficient secure aggregation," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1222–1226.

[82] R. Schlegel, S. Kumar, E. Rosnes *et al.*, "Codedpaddedfl and codedsecagg: Straggler mitigation and secure aggregation in federated learning," *arXiv preprint arXiv:2112.08909*, 2021.

[83] A. R. Elkordy and A. S. Avestimehr, "Heterosag: Secure aggregation with heterogeneous quantization in federated learning," *IEEE Transactions on Communications*, 2022.

[84] I. Ergun, H. U. Sami, and B. Guler, "Sparsified secure aggregation for privacy-preserving federated learning," *arXiv preprint arXiv:2112.12872*, 2021.

[85] J. Cui, C. Chen, T. Ye, and L. Wang, "Practical and light-weight secure aggregation for federated submodel learning," *arXiv preprint arXiv:2111.01432*, 2021.

[86] C. Niu, F. Wu, S. Tang, L. Hua, R. Jia, C. Lv, Z. Wu, and G. Chen, "Secure federated submodel learning," *arXiv preprint arXiv:1911.02254*, 2019.

[87] J. So, B. Güler, and A. S. Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE Journal on Selected Areas in Information Theory*, 2021.

[88] T. Jahani-Nezhad, M. A. Maddah-Ali, S. Li, and G. Caire, "Swiftagg: Communication-efficient and dropout-resistant secure aggregation for federated learning with worst-case security guarantees," *arXiv preprint arXiv:2202.04169*, 2022.

[89] B. Choi, J.-y. Sohn, D.-J. Han, and J. Moon, "Communication-computation efficient secure aggregation for federated learning," *arXiv preprint arXiv:2012.05433*, 2020.

[90] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (poly) logarithmic over-

head," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1253–1269.

[91] K. Mandal, G. Gong, and C. Liu, "Nike-based fast privacy-preserving highdimensional data aggregation for mobile devices," *IEEE T Depend Secure; Technical Report; University of Waterloo: Waterloo, ON, Canada*, pp. 142–149, 2018.

[92] E. Shi, T. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proceedings of the Network and Distributed System Security Symposium, NDSS*, 2011.

[93] Z. Jiang, W. Wang, and Y. Liu, "Flashe: Additively symmetric homomorphic encryption for cross-silo federated learning," *arXiv preprint arXiv:2109.00675*, 2021.

[94] R. Wang, O. Ersoy, H. Zhu, Y. Jin, and K. Liang, "Feverless: Fast and secure vertical federated learning based on xgboost for decentralized labels," 2021.

[95] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," *arXiv preprint arXiv:2106.06639*, 2021.

[96] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E. Ali, B. Guler, and S. Avestimehr, "Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning," *arXiv e-prints*, pp. arXiv–2109, 2021.

[97] J. So, R. E. Ali, B. Guler, J. Jiao, and S. Avestimehr, "Securing secure aggregation: Mitigating multi-round privacy leakage in federated learning," *arXiv preprint arXiv:2106.03328*, 2021.

[98] R. Nasirigerdeh, R. Torkzadehmahani, J. Matschinske, J. Baumbach, D. Rueckert, and G. Kaissis, "Hyfed: A hybrid federated framework for privacy-preserving machine learning," *arXiv preprint arXiv:2105.10545*, 2021.

[99] Z. Liu, S. Chen, J. Ye, J. Fan, H. Li, and X. Li, "Efficient secure aggregation based on shprg for federated learning," *arXiv preprint arXiv:2111.12321*, 2021.

[100] Y. Zhao and H. Sun, "Information theoretic secure aggregation with user dropouts," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 1124–1129.

[101] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multiparty computing," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6178–6186, 2020.

[102] L. Ge, X. He, G. Wang, and J. Yu, "Chain-aafl: Chained adversarial-aware federated learning framework," in *International Conference on Web Information Systems and Applications*. Springer, 2021, pp. 237–248.

[103] Q. Chen, Z. Wang, W. Zhang, and X. Lin, "Ppt: A privacy-preserving global model training protocol for federated learning in p2p networks," *arXiv preprint arXiv:2105.14408*, 2021.

[104] T. Sandholm, S. Mukherjee, and B. A. Huberman, "Safe: Secure aggregation with failover and encryption," *arXiv preprint arXiv:2108.05475*, 2021.

[105] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

[106] Y. Feng, X. Yang, W. Fang, S.-T. Xia, and X. Tang, "Practical and bilateral privacy-preserving federated learning," 2020.

[107] K. Mandal and G. Gong, "Privfl: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 2019, pp. 57–68.

[108] V. Mugunthan, A. Polychroniadou, D. Byrd, and T. H. Balch, "Smpai: Secure multi-party computation for federated learning," in *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

[109] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[110] G. Han, T. Zhang, Y. Zhang, G. Xu, J. Sun, and J. Cao, "Verifiable and privacy preserving federated learning without fully trusted centers," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[111] X. Guo, Z. Liu, J. Li, J. Gao, B. Hou, C. Dong, and T. Baker, "V eri fl: Communication-efficient and fast verifiable aggregation for federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1736–1751, 2020.

[112] C. Hahn, H. Kim, M. Kim, and J. Hur, "Versa: Verifiable secure aggregation for cross-device federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[113] L. Burkhalter, H. Lycklama, A. Viand, N. Küchler, and A. Hithnawi, "Rofl: Attestable robustness for secure federated learning," *arXiv preprint arXiv:2107.03311*, 2021.

[114] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, 2017.

[115] Y. Dong, X. Chen, L. Shen, and D. Wang, "Eastfly: Efficient and secure ternary federated learning," *Computers & Security*, 2020.

[116] F. Chen, P. Li, and T. Miyazaki, "In-network aggregation for privacy-preserving federated learning," in *2021 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 2021, pp. 49–56.

[117] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10 782–10 793, 2020.

[118] X. Zhang, A. Fu, H. Wang, C. Zhou, and Z. Chen, "A privacy-preserving and verifiable federated learning scheme," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[119] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021.

[120] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, J. Joshi, and H. Ludwig, "Fedv: Privacy-preserving federated learning over vertically partitioned data," in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, 2021, pp. 181–192.

[121] F. Tang, W. Wu, J. Liu, H. Wang, and M. Xian, "Privacy-preserving distributed deep learning via homomorphic re-encryption," *Electronics*, vol. 8, no. 4, p. 411, 2019.

[122] W. Yang, B. Liu, C. Lu, and N. Yu, "Privacy preserving on updated parameters in federated learning," in *Proceedings of the ACM Turing Celebration Conference-China*, 2020, pp. 27–31.

[123] D. Xu, S. Yuan, and X. Wu, "Achieving differential privacy in vertically partitioned multiparty learning," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 5474–5483.

[124] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *2020 USENIX Annual Technical Conference, USENIX ATC*, A. Gavrilovska and E. Zadok, Eds., 2020.

[125] M. Asad, A. Moustafa, and T. Ito, "Fedopt: Towards communication efficiency and privacy preservation in federated learning," *Applied Sciences*, vol. 10, no. 8, p. 2864, 2020.

[126] J. Guo, Z. Liu, K.-Y. Lam, J. Zhao, and Y. Chen, "Privacy-enhanced federated learning with weighted aggregation," in *International Symposium on Security and Privacy in Social Networks and Big Data*. Springer, 2021, pp. 93–109.

[127] S. Zhang, Z. Li, Q. Chen, W. Zheng, J. Leng, and M. Guo, "Dubhe: Towards data unbiasedness with homomorphic encryption in federated learning client selection," in *50th International Conference on Parallel Processing*, 2021, pp. 1–10.

[128] A. B. Alexandru and G. J. Pappas, "Private weighted sum aggregation for distributed control systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11 081–11 088, 2020.

[129] D. Stripelis, H. Saleem, T. Ghai, N. Dhinagar, U. Gupta, C. Anastasiou, G. Ver Steeg, S. Ravi, M. Naveed, P. M. Thompson *et al.*, "Secure neuroimaging analysis using federated learning with homomorphic encryption," in *17th International Symposium on Medical Information Processing and Analysis*. SPIE, 2021.

[130] Y. Cheng, Y. Liu, T. Chen, and Q. Yang, "Federated learning for privacy-preserving ai," *Communications of the ACM*, 2020.

[131] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[132] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019.

[133] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[134] J. Zhang, B. Chen, S. Yu, and H. Deng, "Pefl: A privacy-enhanced federated learning scheme for big data analytics," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019.

[135] M. Asad, A. Moustafa, and M. Aslam, "Ceep-fl: A comprehensive approach for communication efficiency and enhanced privacy in federated learning," *Applied Soft Computing*, 2021.

[136] H. Zhu, R. Wang, Y. Jin, and K. Liang, "Pivodl: Privacy-preserving vertical federated learning over distributed labels," *IEEE Transactions on Artificial Intelligence*, 2021.

[137] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM workshop on artificial intelligence and security*, 2019, pp. 1–11.

[138] Y. Liu, Z. Ma, X. Liu, S. Ma, S. Nepal, R. H. Deng, and K. Ren, "Boosting privately: Federated extreme gradient boosting for mobile crowdsensing," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 1–11.

[139] Y. Li, H. Li, G. Xu, X. Huang, and R. Lu, "Efficient privacy-preserving federated learning with unreliable users," *IEEE Internet of Things Journal*, 2021.

[140] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, 2022.

[141] Z. L. Jiang, H. Guo, Y. Pan, Y. Liu, X. Wang, and J. Zhang, "Secure neural network in federated learning with model aggregation under multiple keys," in *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2021, pp. 47–52.

[142] W. Zheng, R. A. Popa, J. E. Gonzalez, and I. Stoica, "Helen: Maliciously secure coopetitive learning for linear models," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[143] H. Zhu, R. Wang, Y. Jin, K. Liang, and J. Ning, "Distributed additive encryption and quantization for privacy preserving federated deep learning," *Neurocomputing*, 2021.

[144] H. Tian, F. Zhang, Y. Shao, and B. Li, "Secure linear aggregation using decentralized threshold additive homomorphic encryption for federated learning," *arXiv preprint arXiv:2111.10753*, 2021.

[145] E. Hosseini and A. Khisti, "Secure aggregation in federated learning via multiparty homomorphic encryption," in *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021, pp. 1–6.

[146] D. Froelicher, J. R. Troncoso-Pastoriza, A. Pyrgelis, S. Sav, J. S. Sousa, J.-P. Bossuat, and J.-P. Hubaux, "Scalable privacy-preserving distributed learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 2, pp. 323–347, 2021.

[147] S. Sav, A. Pyrgelis, J. R. Troncoso-Pastoriza, D. Froelicher, J. Bossuat, J. S. Sousa, and J. Hubaux, "POSEIDON: privacy-preserving federated neural network learning," in *28th Annual Network and Distributed System Security Symposium, NDSS*, 2021.

[148] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 13–23.

[149] D. Wu, M. Pan, Z. Xu, Y. Zhang, and Z. Han, "Towards efficient secure aggregation for model update in federated learning," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[150] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, J. Joshi, and H. Ludwig, "Fedv: Privacy-preserving federated learning over vertically partitioned data," in *AISec@CCS 2021: Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, N. Carlini, A. Demontis, and Y. Chen, Eds., 2021.

[151] D. Boer and S. Kramer, "Secure sum outperforms homomorphic encryption in (current) collaborative deep learning," *arXiv preprint arXiv:2006.02894*, 2020.

[152] E. Sotthiwat, L. Zhen, Z. Li, and C. Zhang, "Partially encrypted multi-party computation for federated learning," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2021, pp. 828–835.

[153] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, "Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning," *arXiv preprint arXiv:2009.11248*, 2020.

[154] Y. Xu, C. Peng, W. Tan, Y. Tian, M. Ma, and K. Niu, "Non-interactive verifiable privacy-preserving federated learning," *Future Generation Computer Systems*, vol. 128, pp. 365–380, 2022.

[155] L. He, S. P. Karimireddy, and M. Jaggi, "Secure byzantine-robust machine learning," *arXiv preprint arXiv:2006.04747*, 2020.

[156] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[157] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," *Advances in Neural Information Processing Systems*, 2018.

[158] H. Chen, H. Li, G. Xu, Y. Zhang, and X. Luo, "Achieving privacy-preserving federated learning with irrelevant updates over e-health applications," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[159] C. Brunetta, G. Tsaloli, B. Liang, G. Banegas, and A. Mitrokotsa, "Non-interactive, secure verifiable aggregation for decentralized, privacy-preserving learning," in *Australasian Conference on Information Security and Privacy*. Springer, 2021, pp. 510–528.

[160] R. Kanagavelu, Z. Li, J. Samsudin, Y. Yang, F. Yang, R. S. M. Goh, M. Cheah, P. Wiwatphonthana, K. Akkarajitsakul, and S. Wang, "Two-phase multi-party computation enabled privacy-preserving federated learning," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020, pp. 410–419.

[161] H. Zhu, R. S. M. Goh, and W.-K. Ng, "Privacy-preserving weighted federated learning within the secret sharing framework," *IEEE Access*, vol. 8, pp. 198 275–198 284, 2020.

[162] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, and J. Zhang, "Vfl: A verifiable federated learning with privacy-preserving for big data in industrial iot," *IEEE Transactions on Industrial Informatics*, 2020.

[163] S. Sharma, C. Xing, Y. Liu, and Y. Kang, "Secure and efficient federated transfer learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2569–2576.

[164] C. Chen, J. Zhou, L. Wang, X. Wu, W. Fang, J. Tan, L. Wang, A. X. Liu, H. Wang, and C. Hong, "When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2652–2662.

[165] F. Zheng, C. Chen, and X. Zheng, "Towards secure and practical machine learning via secret sharing and random permutation," *arXiv preprint arXiv:2108.07463*, 2021.

[166] W. Fang, D. Zhao, J. Tan, C. Chen, C. Yu, L. Wang, L. Wang, J. Zhou, and B. Zhang, "Large-scale secure xgb for vertical federated learning," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

[167] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.

[168] M. Hastings, B. Hemenway, D. Noble, and S. Zdancewic, "Sok: General purpose compilers for secure multi-party computation," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019.

[169] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, 2019.

[170] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, 2020.

[171] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous sgd," *arXiv preprint arXiv:1604.00981*, 2016.

[172] C. Niu, F. Wu, S. Tang, L. Hua, R. Jia, C. Lv, Z. Wu, and G. Chen, "Billion-scale federated learning on mobile clients: A submodel design with tunable privacy," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020.

[173] L. Shi, J. Shu, W. Zhang, and Y. Liu, "Hfl-dp: Hierarchical federated learning with differential privacy," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–7.

[174] J. Zhang, J. Wang, Y. Zhao, and B. Chen, "An efficient federated learning scheme with differential privacy in mobile edge computing," in *International Conference on Machine Learning and Intelligent Communications*. Springer, 2019, pp. 538–550.

[175] C. Wang, C. Ma, M. Li, N. Gao, Y. Zhang, and Z. Shen, "Protecting data privacy in federated learning combining differential privacy and weak encryption," in *International Conference on Science of Cyber Security*. Springer, 2021, pp. 95–109.

[176] R. Liu, Y. Cao, M. Yoshikawa, and H. Chen, "Fedsel: Federated sgd under local differential privacy with top-k dimension selection," in *International Conference on Database Systems for Advanced Applications*. Springer, 2020, pp. 485–501.

[177] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 61–66.

[178] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, and H. V. Poor, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Transactions on Mobile Computing*, 2021.

[179] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[180] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[181] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[182] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[183] Z. Chuanxin, S. Yi, and W. Degang, "Federated learning with gaussian differential privacy," in *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*, 2020, pp. 296–301.

[184] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[185] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-iid data in aiot," *IEEE Transactions on Industrial Informatics*, 2021.

[186] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, 2019.

[187] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2587–2596.

[188] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Generation Computer Systems*, 2022.

[189] L. Yin, J. Feng, H. Xun, Z. Sun, and X. Cheng, "A privacy-preserving federated learning for multiparty data sharing in social iots," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2706–2718, 2021.

[190] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[191] B. Ghazi, R. Pagh, and A. Velingker, "Scalable and differentially private distributed aggregation in the shuffled model," *arXiv preprint arXiv:1906.08320*, 2019.

[192] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2521–2529.

[193] L. Sun, J. Qian, and X. Chen, "Ldp-fl: Practical private aggregation in federated learning with local differential privacy," *arXiv preprint arXiv:2007.15789*, 2020.

[194] Y. Liu, R. Zhao, J. Kang, A. Yassine, D. Niyato, and J. Peng, "Towards communication-efficient and attack-resistant federated edge learning for industrial internet of things," *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 3, pp. 1–22, 2021.

[195] N. Agarwal, P. Kairouz, and Z. Liu, "The skellam mechanism for differentially private federated learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[196] R. Hu, Y. Guo, and Y. Gong, "Concentrated differentially private federated learning with performance analysis," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 276–289, 2021.

[197] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, 2019.

[198] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, 2020.

[199] S. Kim, "Incentive design and differential privacy based federated learning: A mechanism design perspective," *IEEE Access*, vol. 8, pp. 187 317–187 325, 2020.

[200] X. Chen, J. Ji, C. Luo, W. Liao, and P. Li, "When machine learning meets blockchain: A decentralized, privacy-preserving and secure design," in *2018 IEEE international conference on big data (big data)*. IEEE, 2018, pp. 1178–1187.

[201] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. Abd El-Latif, "A secure federated learning framework for 5g networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 24–31, 2020.

[202] T. Rückel, J. Sedlmeir, and P. Hofmann, "Fairness, integrity, and privacy in a scalable blockchain-based federated learning system," *Computer Networks*, vol. 202, p. 108621, 2022.

[203] S. Bhagavan, M. Gharibi, and P. Rao, "Fedsmarteum: Secure federated matrix factorization using smart contracts for multi-cloud supply chain," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 4054–4063.

[204] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in iiot," *IEEE Transactions on Industrial Informatics*, 2021.

[205] X. Zhu and H. Li, "Privacy-preserving decentralized federated deep learning," in *ACM Turing Award Celebration Conference-China (ACM TURC 2021)*, 2021, pp. 33–38.

[206] Z. Li, J. Liu, J. Hao, H. Wang, and M. Xian, "Crowdsfl: a secure crowd computing framework based on blockchain and federated learning," *Electronics*, vol. 9, no. 5, p. 773, 2020.

[207] C. Jiang, C. Xu, and Y. Zhang, "Pflm: Privacy-preserving federated learning with membership proof," *Information Sciences*, 2021.

[208] C. Fang, Y. Guo, J. Ma, H. Xie, and Y. Wang, "A privacy-preserving and verifiable federated learning method based on blockchain," *Computer Communications*, 2022.

[209] M. Qi, Z. Wang, F. Wu, R. Hanson, S. Chen, Y. Xiang, and L. Zhu, "A blockchain-enabled federated learning model for privacy preservation: System design," in *Australasian Conference on Information Security and Privacy*. Springer, 2021, pp. 473–489.

[210] U. Majeed, L. U. Khan, A. Yousafzai, Z. Han, B. J. Park, and C. S. Hong, "St-bfl: A structured transparency empowered cross-silo federated learning on the blockchain framework," *IEEE Access*, vol. 9, pp. 155 634–155 650, 2021.

[211] N. Wang, W. Yang, Z. Guan, X. Du, and M. Guizani, "Bpfl: A blockchain based privacy-preserving federated learning scheme," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[212] J. Li, Y. Shao, K. Wei, M. Ding, C. Ma, L. Shi, Z. Han, and V. Poor, "Blockchain assisted decentralized federated learning (blade-fl): Performance analysis and resource allocation," *IEEE Transactions on Parallel and Distributed Systems*, 2021.

[213] Z. Liu, N. C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "A survey on blockchain: A game theoretical perspective," *IEEE Access*, 2019.

[214] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "Sear: Secure and efficient aggregation for byzantine-robust federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[215] F. Mo and H. Haddadi, "Efficient and private federated learning using tee," in *Proc. EuroSys Conf., Dresden, Germany*, 2019.

[216] H. Hashemi, Y. Wang, C. Guo, and M. Annavaram, "Byzantine-robust and privacy-preserving framework for fedml," 2021.

[217] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: what it is, and what it is not," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, pp. 57–64.

[218] Y. Zhang, Z. Wang, J. Cao, R. Hou, and D. Meng, "Shufflefl: gradient-preserving federated learning using trusted execution environment," in *Proceedings of the 18th ACM International Conference on Computing Frontiers*, 2021, pp. 161–168.

[219] A. Boutet, T. Lebrun, J. Aalmoes, and A. Baud, "Mixnn: Protection of federated learning against inference attacks by mixing neural network layers," *arXiv preprint arXiv:2109.12550*, 2021.

[220] F. Mo, A. S. Shamsabadi, K. Katevas, S. Demetriou, I. Leontiadis, A. Cavallaro, and H. Haddadi, "Darknetz: towards model privacy at the edge using trusted execution environments," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 161–174.

[221] E. Kuznetsov, Y. Chen, and M. Zhao, "Securefl: Privacy preserving federated learning with sgx and trustzone," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2021.

[222] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "Ppfl: privacy-preserving federated learning with trusted execution environments," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 94–108.

[223] W. Zhang and T. Muhr, "Tee-based selective testing of local workers in federated learning systems," in *2021 18th International Conference on Privacy, Security and Trust (PST)*. IEEE, 2021.

[224] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Information Sciences*, 2020.

[225] K. Bonawitz, H. Eichner, W. Grieskamp *et al.*, "Tensorflow federated: machine learning on decentralized data.(2020)."

[226] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *Journal of Machine Learning Research*, vol. 22, no. 226, pp. 1–6, 2021.

[227] E. Hauck and J. Loss, "Efficient and universally composable protocols for oblivious transfer from the cdh assumption," *Cryptology ePrint Archive*, 2017.

[228] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*. IEEE, 1987, pp. 427–438.

[229] "Baidu PaddlePaddle Releases 21 New Capabilities to Accelerate Industry-Grade Model Development." Available: http://research.baidu.com/Blog/index-view?id=126.

[230] K. He, L. Yang, J. Hong, J. Jiang, J. Wu, X. Dong, and Z. Liang, "Privc—a framework for efficient secure two-party computation," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2019, pp. 394–407.

[231] "PySyft." Available: https://github.com/OpenMined/PySyft.

[232] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[233] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[234] V. Mugunthan, A. Peraire-Bueno, and L. Kagal, "Privacyfl: A simulator for privacy-preserving and secure federated learning," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 3085–3092.

[235] "We Research and Build Artificial Intelligence Technology and Services." Available: https://sherpa.ai/.

[236] "Hive." URL https://www.intel.com/content/customer-spotlight/stories/ping-an-sgx-customer-story.html.

[237] B. Liu, C. Tan, J. Wang, T. Zeng, H. Shan, H. Yao, H. Huang, P. Dai, L. Bo, and Y. Chen, "Fedlearn-algo: A flexible open-source privacy-preserving machine learning platform," *arXiv preprint arXiv:2107.04129*, 2021.

[238] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[239] "Clara Train SDK Documentation." Available: https://docs.nvidia.com/clara/tlt-mi/clara-train-sdk-v3.1/.

[240] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn *et al.*, "Ibm federated learning: an enterprise framework white paper v0. 1," *arXiv preprint arXiv:2007.10987*, 2020.

[241] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

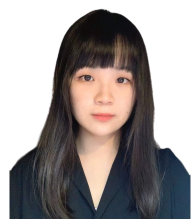
[242] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011.

[243] S. Andreina, G. A. Marson, H. Möllering, and G. Karame, "Baffle: Backdoor detection via feedback-based federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 852–863.

Ziyao Liu received his B.E. degree from the school of Electronics Information Engineering, Zhengzhou University, Zhengzhou, China, in 2015, and his M.S. degree from Beijing Institute of Technology, Beijing, China, in 2018. He is currently working towards a Ph.D. degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include privacy-preserving machine learning, multi-party computation, and applied cryptography.

Jiale Guo received her B.S. from the School of Mathematics, Shandon University, Jinan, China, in 2017. She is currently pursuing a Ph.D. degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Her research interests include Privacy-Preserving machine learning and Cybersecurity.

Wenzhuo Yang received the Bachelor's degree in Measuring and Controlling Technologies and Instruments from Beijing University of Posts and Telecommunications, Beijing, China, in 2016. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Her current research interests are in the area of Privacy-Preserving Machine Learning, IoT Security, Intrusion Detection, and Cyber Threat Intelligence Analysis.



Jiani Fan received the Bachelor's degree in information systems from Singapore Management University in 2020. She is currently pursuing the Ph.D. degree in computer science at Nanyang Technological University. Her research interests include IoT Security, Cybersecurity, and Internet of Vehicles.

Kwok-Yan Lam (Senior Member, IEEE) received his B.Sc. degree (1st Class Hons.) from University of London, in 1987, and Ph.D. degree from University of Cambridge, in 1990. He is the Associate Vice President (Strategy and Partnerships) and Professor in the School of Computer Science and Engineering at the Nanyang Technological University, Singapore. He is currently also the Director of the Strategic Centre for Research in Privacy-Preserving Technologies and Systems (SCRiPTS). From August 2020, he is on part-time secondment to the INTERPOL as a Consultant at Cyber and New Technology Innovation. Prior to joining NTU, he has been a Professor of the Tsinghua University, PR China (2002–2010) and a faculty member of the National University of Singapore and the University of London since 1990. He was a Visiting Scientist at the Isaac Newton Institute, Cambridge University, and a Visiting Professor at the European Institute for Systems Security. In 1998, he received the Singapore Foundation Award from the Japanese Chamber of Commerce and Industry in recognition of his research and development achievement in information security in Singapore. His research interests include Distributed Systems, Intelligent Systems, IoT Security, Distributed Protocols for Blockchain, Homeland Security and Cybersecurity.

Jun Zhao (S'10-M'15) is currently an Assistant Professor in the School of Computer Science and Engineering (SCSE) at Nanyang Technological University (NTU), Singapore. He received a Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, USA, in May 2015, and a bachelor's degree in Information Engineering from Shanghai Jiao Tong University, China, in June 2010. One of his papers was a finalist for the best student paper award in IEEE International Symposium on Information Theory (ISIT) 2014. His research interests include A.I. and data science, security and privacy, control and learning in communications and networks.