The ISC Int'l Journal of Information Security

November 2023, Volume 15, Number 3 (pp. 67–76)

http://www.isecure-journal.org

Presented at the ISCISC'2023 in Tehran, Iran.

Private Federated Learning: An Adversarial Sanitizing Perspective **

Mojtaba Shirinjani [1,*], Siavash Ahmadi [2], Taraneh Eghlidos [2], and Mohammad R. Aref [1]

[1] *Information Systems and Security Lab, EE Department, Sharif University of Technology, Tehran, Iran*

[2] *Electronics Research Institute, Sharif University of Technology, Tehran, Iran*

ARTICLE INFO.

Keywords:
Byzantine-resilience, Differential Privacy, Federated Learning, Homomorphic Encryption

Type:
Research Article

doi:
10.22042/isecure.2023.182211

ABSTRACT

Large-scale data collection is challenging in alternative centralized learning as privacy concerns or prohibitive policies may rise. As a solution, Federated Learning (FL) is proposed wherein data owners, called participants, can train a common model collaboratively while their privacy is preserved. However, recent attacks, namely Membership Inference Attacks (MIA) or Poisoning Attacks (PA), can threaten the privacy and performance in FL systems. This paper develops an innovative Adversarial-Resilient Privacy-preserving Scheme (ARPS) for FL to cope with preceding threats using differential privacy and cryptography. Our experiments display that ARPS can establish a private model with high accuracy out¬performing state-of-the-art approaches. To the best of our knowledge, this work is the only scheme providing privacy protection beyond any output models in conjunction with Byzantine resiliency without sacrificing accuracy and efficiency.

1 Introduction

The performance of Machine Learning (ML) models greatly relies on the quantity of training data available. Traditional ML algorithms typically involve a single entity conducting centralized learning which can in-crease computational complexity. Further, getting access to a large dataset may be challenging. Consider a case in which the WHO intends to train a model assisting the cancer diagnosis based on the medical records of some countries. However, privacy concerns and government policies motivate these countries to avoid data sharing. Besides, some real-time applications, such as traffic flow prediction, require updates in near real-time. However, centralized learning methods can often prove excessively time-consuming for such purposes. Federated learning (FL) is a strategy for overcoming the barriers to data sharing, enabling participants to train a common model collaboratively without revealing their datasets [1]. Moreover, the distributed nature of FL can take advantage of parallelization to accelerate the training process and leads to faster convergence. In this strategy, local models trained over user-side datasets are aggregated in a central server as the

* Corresponding author.

**The ISCISC'2023 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: mojtaba.shirinjani@ee.sharif.edu, s.ahmadi@sharif.edu, teghlidos@sharif.edu, aref@sharif.edu

ISSN: 2008-2045

global model. Then, participants update their local models using the global model and share them with the server. This iterative process continues until a convergence criterion is met.

However, ML models can memorize the details related to individual examples rather than encode the dataset's general patterns. A resolute adversary can exploit this breach to mount a Membership Inference Attack (MIA) [2] to extract information about the training data from raw models sent to/from the server, causing critical privacy issues. Further, FL schemes using local differential privacy (DP) [3] to address privacy problems can lead to low-performance models due to adding too much noise during the learning process. On the other hand, some adversarial users aim to perturb the learning procedure via manipulating their local models during protocol execution [4]. Concerning Poisoning attacks (PA), Byzantine-robust schemes [4, 5] have been proposed wherein an outlier detection algorithm is used to remove potentially malicious participants. Accordingly, these approaches are insufficient to provide (i) privacy protection against information leakage from the local or global models and (ii) a communication efficient algorithm against Byzantine faults. To address the above challenges, we propose an Adversarial-Resilient Privacy-preserving Scheme (ARPS) for federated learning, which can efficiently identify poisoning incidents during the training process, while protecting the privacy. Particularly, the contributions of this paper are threefold, as follows:

- We propose ARPS as a privacy-preserving FL scheme that tolerates Byzantine faults under homomorphic encryption technology while improving accuracy.
- We develop a novel similarity-based outlier detection algorithm (**Sanitize**) to identify the malicious models.
- We illustrate how our approach can be implemented to train ML models, specifically convolutional neural networks, and present empirical assessments.

2 Related Work

2.1 Defending Against Poisoning Attacks

In the Byzantine context, aggregation rules such as Krum [6], and Median [7], can be used to enhance FL against malfunctioning. However, these defenses may collapse, facing countermeasures such as optimized local model poisoning attacks [4]. A correlation-based defense technology is presented in [8] that can detect poisoning attacks. However, the correlation between the gradients is revealed in this approach, which may lead to privacy violations. The work in [9] effectively defend against poisoning attacks by leveraging contribution similarity for multiple threat actors. However, it may not be as effective against single attackers. BREA [5] leverages verifiable secret sharing to protect the privacy of individual participants. However, cubic communication overhead is its main limitation. In [10], a feedback-based detection method called BaFFLe is utilized to combat backdoor attacks. However, this scheme suffers from a low convergence rate. Recently, [11] has presented a hybrid system using zero-knowledge proof and RSA optimization algorithm to provide privacy protection with Byzantine robustness. However, this scheme is vulnerable to inference beyond the global model. An FL-based intrusion detection system is introduced in [12] that proposes a two-stage defense algorithm, DPA-FL, to mitigate the impact of backdoor and label-flipping attacks. The authors in [13] believe that the non-IID nature of user data can increase the false positive rate for defense algorithms. Initially, the local models are partitioned into multiple clusters using Microaggregation to address this. Subsequently, models with a distance exceeding a specified threshold from the cluster centroids are identified as outliers.

2.2 Avoiding Privacy Violation

In the non-Byzantine setting, secure aggregation can be tackled with additive masking [14]. In this method, participants upload a masked version of their local models to the server. Due to the additive property of masking, the server can add up the masked models to disclose the aggregated models. However, this approach may fail due to participant model drops, and it costs a quadratic communication overhead. Cryptographic approaches in [5, 15] propose techniques to aggregate models using their encrypted data rather than raw ones. Homomorphic encryption and verifiable secret sharing are the two main cryptographic tools used in the aggregation process due to their additive property. HybridAlpha [16] is another private approach employing an SMC protocol based on functional encryption. Although this approach allows for high privacy guarantees without performance loss, it is vulnerable to inference beyond the global model. DP is another line of work wherein differentially private schemes [17] can raise against inference attacks.

The rest of this paper is organized as follows. In Section 3, we outline the preliminaries used to design ARPS. We then, in Section 4, discuss ARPS in detail and allocate an algorithmic description of its functionality. In Section 5, we describe system settings and empirical results. Finally, we conclude the paper in Section 6.

3 Preliminaries

In this section, we outline the primary components required to construct our scheme, including differential privacy, and homomorphic encryption.

3.1 Differential Privacy

Differential privacy [18] is a probabilistic mechanism that can be used to randomize a response given to a database query, in which the inclusion or exclusion of any single record in the dataset results in statistically meaningless changes to the outcome. Roughly speaking, DP provides plausible deniability for individuals to guarantee their privacy against inference beyond the learning models. The formal definition for DP is as follows.

Definition 1. A randomized mechanism $\mathscr{M}$ with domain $\mathscr{D}$ and range $\mathscr{R}$ provides (ϵ, δ)-differential privacy if for any two adjacent datasets $\mathscr{D}_1, \mathscr{D}_2 \in \mathscr{D}$ that differ in only a single record, and for all outcome subsets $\mathscr{S} \subseteq \mathscr{R}$, we have:

$$P_r(\mathscr{M}(\mathscr{D}_1) \in \mathscr{S}) \leq e^{\epsilon} P_r(\mathscr{M}(\mathscr{D}_2) \in \mathscr{S}) + \delta \quad (1)$$

The parameter δ represents the probability that ϵ-differential privacy fails to protect privacy. The parameter ϵ is the privacy budget that controls the privacy loss of mechanism $\mathscr{M}$ such that the larger values it takes, the more privacy loss results. DP is applied to an algorithm's output using noise injection with the magnitude proportional to the sensitivity of the output. Sensitivity is a measure criterion to track the maximum change at the output on account of a single-record inclusion in the dataset. A randomization mechanism can be generated using Gaussian noise as the prevalent mechanism in the matter of learning and defined by:

$$\mathscr{M}(\mathscr{D}) = f(\mathscr{D}) + \mathscr{N}(0, \Delta f^2 \sigma^2), \quad (2)$$

where $\mathscr{N}(0, \Delta f^2 \sigma^2)$ stands for normal distribution with zero mean and variance of $\Delta f^2 \sigma^2$. The Gaussian mechanism applied to function f of sensitivity Δf satisfies (ϵ, δ)-differential privacy if $\delta \geq 1.25 exp(-(\sigma\epsilon)^2/2)$ and $0 < \epsilon < 1$ [18].

3.2 Homomorphic Encryption

Homomorphic encryption is a public-key cryptographic tool that enables users to perform computations over encrypted data without requiring a private key. Additive homomorphic encryption schemes, like Paillier, provide the following properties:

$$Enc_{pbk}(P_1) * Enc_{pbk}(P_2) = Enc_{pbk}(P_1 + P_2) \quad (3)$$
$$Enc_{pbk}(P) * *r = Enc_{pbk}(r.P), \quad (4)$$

wherein P_i and pbk denote the plaintext and public-key respectively. Operation $*$ serves as the multiplication and, $**$ serves as the exponentiation. Suchlike schemes are functional when untrusted parties carry out the computations over encrypted data leading to the privacy protection of the data owner. Paillier scheme [19] is an additive homomorphic public-key cryptosystem that provides probabilistic encryption based on computations in group $Z^*_{n^2}$, where n is an RSA modulus. The security of this scheme is owing to the difficulty of computing the n^{th} residue classes problem.

4 Proposed Approach

In this section, we present the details required to organize ARPS as a framework for private and secure federated learning against inference attacks and Byzantine faults.

4.1 System Model

Our system consists of four fundamental entities:

- **Key Generation Center (KGC):** The KGC is a trusted entity responsible for the distribution and administration of all public and private keys (pbk, sk).
- **Participants:** Also referred to as users, the participants $\mathscr{P} = \{\mathscr{P}_1, \mathscr{P}_2, ..., \mathscr{P}_N\}$ participate in the training of a shared model under the coordination of the Aggregator. To preserve privacy, each user trains the model locally using their private data from $\mathscr{D} = \{\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_N\}$ on their respective devices. We recall that the servers are assumed to be untrusted, so DP is employed locally by each user. The encrypted gradients are subsequently uploaded to the Aggregator. Furthermore, it is assumed that the data held by each user is independent and identically distributed (IID).
- **Aggregator:** The Aggregator receives user gradients and performs aggregation, typically through averaging, to achieve an optimized global model. Additionally, the Aggregator is tasked with detecting poisoning attacks with the assistance of the Sanitizer.
- **Sanitizer:** The Sanitizer assists the Aggregator in identifying malicious users and holds a private-public key pair (pbk, sk) generated by the KGC for data encryption and decryption.

4.2 Threat Model

In our threat model, the Aggregator and Sanitizer are honest but curious servers. Such a server follows the protocol instructions properly but attempts to

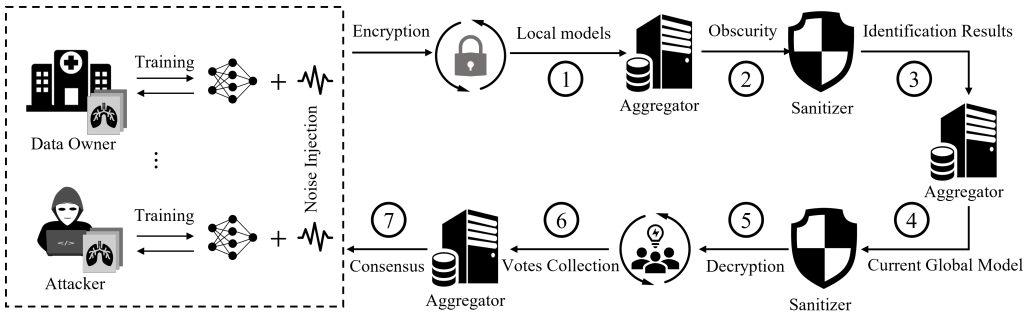

Figure 1. High-level architecture of proposed FL approach

extract additional information. Furthermore, a group of users up to size a can collude with each other, called attackers, and in contrast to the servers, they can circumvent the learning protocol. Gaussian and label-flipping attacks are well-known attacks that attackers can use to craft their local models. In the Gaussian attack, samples from Gaussian distribution with mean zero and a specific amount of variance are substituted for local model parameters. In contrast, a label-flipping attack is a data poisoning attack, wherein adversaries corrupt some data samples by maliciously changing their labels before training. Our threat model also considers the outsiders who monitor the protocol execution and ensures they learn nothing about participants' sensitive information. The communications between the servers and the participants are secured using Paillier cryptosystem. The integrity and authenticity of transmitted messages are supplied via secure channels between the servers and the participants.

4.3 Scheme Design

We employ a novel FL scheme inspired by cryptography, and DP that addresses MIA and PA simultaneously as two dominant attacks in the federated learning area. Initially, Aggregator and participants agree on the number of adversaries a, the number of honest participants $\hbar$, the learning algorithm g, the dataset structure, and the privacy loss ϵ offline. DP-SGD [20], a deep learning algorithm that modifies the minibatch stochastic optimization process to make it differentially private, is served as g. Further, N participants wish to collaboratively train a neural network model in a private fashion. To accomplish these goals, we depict our scheme in the following steps:

(0) The Aggregator broadcasts the initial version of the global model parameters to the participants.

(1) Each participant uses the global model and its respective dataset to update its local model. This is done through the *train˙local˙model* function in Pseudocode 1, as described in [20]. During the learning process, participants apply a DP mechanism to add a proper amount of noise to their models due to privacy budget, number of honest participants, number of adversaries, and sensitivity of g. Afterward, the randomized models are encrypted using Paillier with *pbk* and conveyed to the Aggregator.

(2) Upon receiving the encrypted local models, the Aggregator obscures them by multiplying them

Pseudocode 1 ARPS

Input: Privacy guarantee ϵ; set of participants $\mathcal{P}$; number of honest participants $\hbar$; number of rounds T; learning rate η; sampling probability μ; loss function $\mathcal{L}$; Aggregator $\mathcal{A}$; Sanitizer $\mathcal{S}$; Key generation center **KGC**.
Output: Global model θ^T.
initializing $\theta^(0))$ with random values, KGC generates (pbk, sk) for $\mathcal{S}$.
for $t \in 1, ..., T$ **do**
  for $p \in \mathcal{P}$ **do**
    download the golbal model $\theta^{(t)}$ from $\mathcal{A}$
    compute $\theta_p^{(t)} = train˙local˙model(\mathcal{D}_p, \mathcal{N}(0, \Delta g^2 \sigma^2/\hbar), \eta, \mu, \mathcal{L})$
    send $E_p^{(t)} \leftarrow Enc(\theta_p^{(t)}, pbk)$ to $\mathcal{A}$
  end for
  $\mathcal{A}$ computes $\mathcal{P}_{honest} = Sanitize(\{E_p^{(t)}\}_{p=1}^N, \mathcal{P}, \hbar)$
  $\mathcal{A}$ sends aggregate $E^{(t)} = \prod_{p \in \mathcal{P}_{honest}} E_p^{(t)}$ to $\mathcal{S}$
  $\mathcal{S}$ calculates $\tilde{\theta}^{(t)} = \sum_{p \in \mathcal{P}_{honest}} \theta_p^{(t)} = Dec(E^{(t)}, sk)$
  $\mathcal{S}$ sends $\theta^{(t)} = \tilde{\theta}^{(t)}.|\mathcal{P}_{honest}|^{-1}$ to $\mathcal{A}$ and $\mathcal{P}$
  for $p \in \mathcal{P}_{decryption}$ **do**
    if accuracy$(\theta^{(t)}) \geq$ **accuracy**$(\theta_p^{(t)})$ **then**
      $vote_p = 1$, else: $vote_p = 0$
    end if
    send $vote_p$ to $\mathcal{A}$
  end for
  if $\sum_{p \in \mathcal{P}_{honest}} vote_p \geq 0.7|\mathcal{P}_{honest}|$ **then**
    $\theta^{(t)} = \theta^{(t)}$, else: $\theta^{(t)} = \theta^{(t-1)}$
  end if
end for

Pseudocode 2 Sanitize

Input: $\{E_p^{(t)}\}_{p=1}^N; \mathscr{P}; \hbar$
Output: Honest Clients $\mathscr{P}_{honest}$
$\mathscr{A}$ selects $q \in_R Z^+$
$\mathscr{A}$ sends $\{E_p^{(t)} * *q\}_{p=1}^N$ to $\mathscr{S}$
$\mathscr{S}$ decrypts $\{\theta_{p,blind}^{(t)} * *q\}_{p=1}^N = \{q * \theta_p^{(t)}\}_{p=1}^N = \{Dec(E_p^{(t)} * *q, sk)\}_{p=1}^N$
$\mathscr{S}$ performs $\{score_p^{blind}\}_{p=1}^N = multiKrum(\{\theta_{p,blind}^{(t)}\}_{p=1}^N)$
$\mathscr{S}$ performs $\hat{\theta}_{blind} = Median(\{\theta_{p,blind}^{(t)}\}_{p=1}^N)$
$\mathscr{S}$ obtains $\{similarity_p\}_{p=1}^N = CosineSimilarity(\{\theta_{p,blind}^{(t)}\}_{p=1}^N, \hat{\theta}_{blind})$
$\mathscr{S}$ sends $\{score_p^{blind}\}_{p=1}^N$ and $\{similarity_p\}_{p=1}^N$ to $\mathscr{A}$
$\mathscr{A}$ performs $\{score_p\}_{p=1}^N = \{q^{-1}.score_p^{blind}\}_{p=1}^N$
Aggregator runs $\arg\max y$ and $\arg\min x$ functions for $\hbar$ times
$\mathscr{X}.append(\arg\min_p\{score_p\}_{p=1}^N)$
$\mathscr{Y}.append(\arg\max_p\{similarity_p\}_{p=1}^N)$
$\mathscr{P}_{honest} = \mathscr{X} \bigcap \mathscr{Y}$

with a random non-zero integer q. This is realized by Paillier's additive homomorphism property. Then, the Aggregator sends the obscured local models to the Sanitizer.

(3) In order to identify malicious models, the Sanitizer needs to decrypt the encrypted obscured local models using sk. First, as described in [6], it runs the multi-Krum algorithm to calculate the similarity scores of each obscured model. In the next stage, it computes the cosine similarity between each obscured model and the coordinate-wise median of the obscured local models. Finally, it provides the results of both stages to the Aggregator. By calling **Sanitize** in Pseudocode 2, we describe the details for sanitizing the local models.

(4) In contrast to cosine similarity, that obscurity does not affect its results; a rescaling operation must be applied as follows to complete the multi-Krum algorithm:

$$score_p = (score_p^{blind}).q^{-1} for\ p \in [N] \quad (5)$$

The Aggregator picks up $\hbar$ models with minimum scores and $\hbar$ models with maximum cosine similarity separately. Then, to intensify the **Sanitize**, it averages the joint models of both methods and delivers the result to the Sanitizer.

(5) Before broadcasting the received result as the global model $\theta^{(t)}$ to the participants and the Aggregator, the Sanitizer decrypts it using sk.

(6) The Aggregator employs a consensus mechanism to reduce the likelihood of bypassing the **Sanitize**. To aim this, participants evaluate the accuracy of the current global model $\theta^{(t)}$ using their test dataset. If the accuracy has increased compared to the previous round, the participant submits 1 as a positive vote to Aggregator. Otherwise, it submits 0 as a negative vote.

(7) The Aggregator only adopts the votes from honest users, previously identified by **Sanitize**, and if more than 70% of them are positive, then $\theta^{(t)}$ is permitted to be used as the current global model. Otherwise, $\theta^{(t-1)}$ is downloaded by users as the current global model.

This iterative procedure through step 1 up to step 7 is terminated when the global model converges or meets a bound for the maximum number of rounds. This scenario is illustrated in Figure 1, and details are outlined in Pseudocode 1. It is worth noting that we perform robust aggregation under ciphertext entirely. The only non-cipher data revealed to the public are the global models, which are under DP protection.

4.4 Noise Generation

Applying DP to federated learning in a traditional fashion is performed as follows. Every participant employs the Gaussian mechanism to add noise to its trained model with the distribution of $\mathcal{N}(0, \Delta g^2\sigma^2)$, where Δg denotes the sensitivity of g and σ indicates the noise parameter. In this setting, every single model is differentially private, making the encryption unnecessary for privacy protection. However, the overall noise generated at the Aggregator after the aggregation process will be $\sum_{i=1}^N \mathcal{N}(0, \Delta g^2\sigma^2)$ which equals $\mathcal{N}(0, \Delta g^2 N\sigma^2)$ due to the additive property of Gaussian noise. Hence the global model satisfies $\epsilon/\sqrt{N}$-DP, resulting in meaningful accuracy degradation after multiple rounds of training with a fixed N. We know that DP is immune against post-processing. Hence, by using averaging as aggregation rule, we can reduce the sensitivity. We can go further and consciously decrease the quantity of variance by a factor of $\hbar$. This helps to proceed the training process for more rounds without accuracy drop compared to traditional approaches. In this case, we have directly reduced the variance itself, so the privacy loss will be $\hbar$ times higher than normal situation, which contradicts the ϵ-DP guarantee. Therefore, to provide security for local models, Paillier is leveraged. This strategy leads to an aggregation value with overall noise of:

$$aggregation\ noise = \mathcal{N}(0, (\Delta g/N)^2 N/\hbar\sigma^2) \quad (6)$$

as $N/\hbar$ is strictly greater than one, we have reached a near optimal tradeoff, wherein differential privacy is preserved while at the same time offering significant increments of accuracy.

Remark 1. Although the local models do not con-

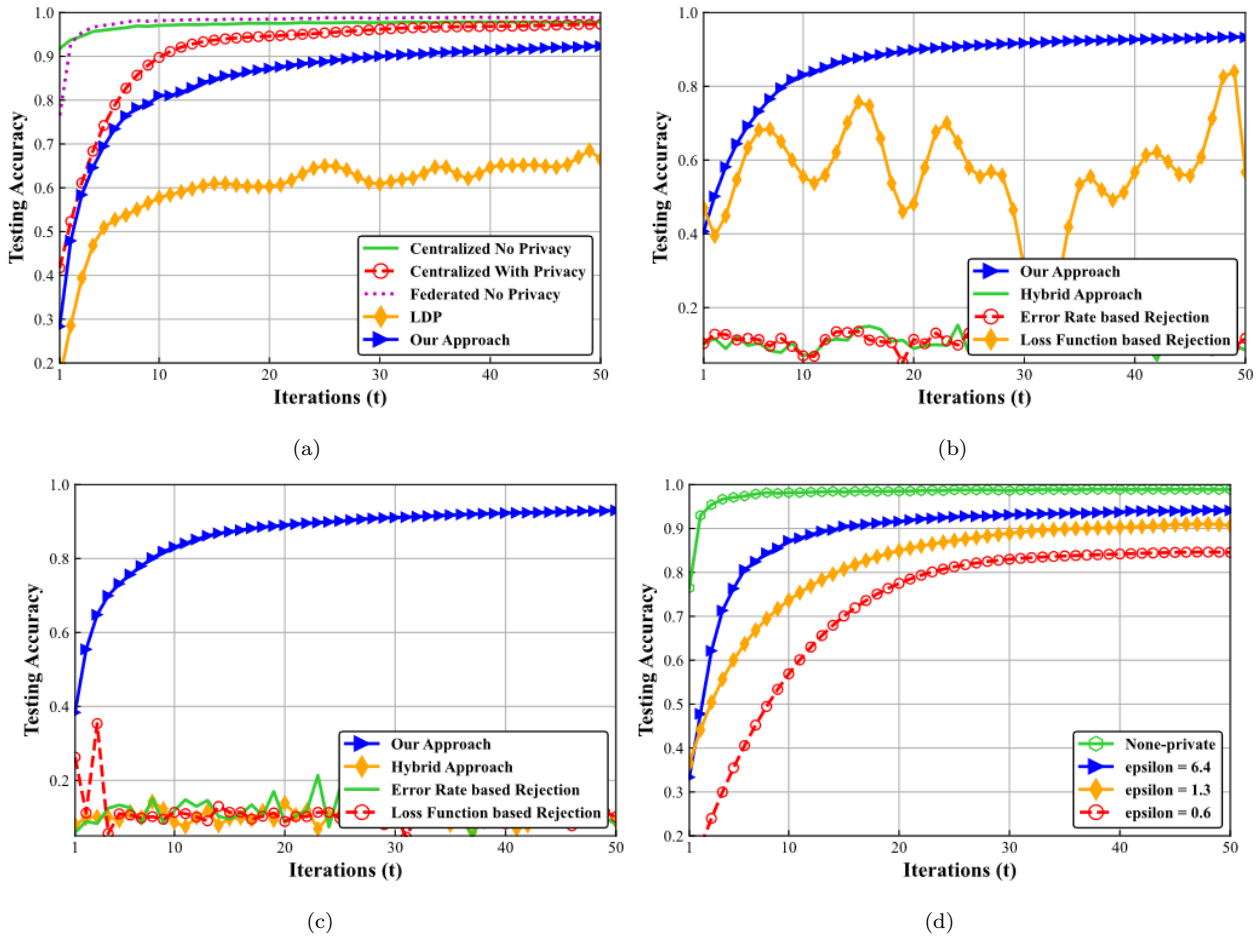

Figure 2. CNN training with MNIST dataset under $\sigma = 8$, $(\epsilon, \delta) = (0.8, 10^{-5})$ for 20 participants. (a) none-Byzantine setting; (b) Under label-flipping attack; (c) Under Gaussian attack; (d) none-Byzantine setting for $\epsilon \in \{0.6, 1.3, 6.4\}$

tain enough noise to ensure differential privacy and thus require encryption, as per Equation (6), all the global models and the final model in our system are protected under (ϵ, δ)-DP. As a result, they can be decrypted and shared publicly.

Remark 2. As local models are encrypted and global models are under (ϵ, δ)-DP protection, no information more than ϵ can be extracted across our system. So, ARPS is private against MIA.

5 Experimental Evaluation

In this section, we empirically evaluate the performance of the proposed ARPS using the convolutional neural networks (CNN) on the popular dataset MNIST used as a benchmark in the machine learning literature.

5.1 Experimental Setup

The MNIST dataset of handwritten digits has a training set of 60,000 samples and a test set of 10,000 samples. Our baseline model is a feedforward neural network (CNN) with a single hidden layer containing 32 hidden units. We use Keras with Tensorflow backend in python to implement neural network architecture. For private optimizing of model parameters and gradients by SGD, we use the Tensorflow privacy library. The phe library is employed to encrypt model parameters, which consist of real numbers, using the Paillier cryptosystem. We conduct experiments by adjusting the number of participants $N = 20$, the number of rounds $T = 50$, the privacy budget $\epsilon \in \{0.6, 1.3, 6.4\}$, and the type of poisoning attack. For the differentially private SGD optimizer of networks, we set the epoch to 1, the learning rate to 0.18, batch size to 150, the norm clipping to 1.5, and the number of micro batches to 150. We compare our results with the six baseline scenarios below to show how our approach can stand out in Byzantine and non-Byzantine settings. Each scenario can be treated as an individual class, with various approaches categorized accordingly.

(1) None-private Centralized Learning. In this scenario, a single party holds the entire dataset, and no privacy mechanisms are applied during

Table 1. Comparison of state-of-the-art Federated Learning approaches

	Security					Efficiency		
FL Approaches	Byzantine Robustness	SMC	Privacy	Source of Information Leakage	Accuracy	No. of Client-side Computations per Iteration	Communication Overhead	Clients' Interaction
[15]	✗	HE*	✓	✗	> 0.9	O(1)	O(N)	✗
[16]	✗	FE‡	✓	✗	> 0.9	O(1)	O(N)	✗
[3]	✗	✗	✓	✗	< 0.9§	O(1)	O(N)	✗
[4]	✓	✗	✗	local-global models	> 0.9	O(1)	O(N)	✗
[5]	✓	SS†	✗	global model	> 0.9	O(N^2)	O(N^2)	✓
[21]	✗	SS	✗	global model	> 0.9	O(1)	O($NlogN$)	✓
[10]	✓	HE-ZKP**	✗	global model	> 0.9	O(1)	O(N)	✗
[8]	✓	HE	✗	global model	> 0.9	O(1)	O(N)	✗
Our Approcah	✓	HE	✓	✗	> 0.9	O(1)	O(N)	✗

∗ Denotes homomorphic encryption.
† Denotes secret sharing.
∗∗ Denotes zero-knowledge proof.
‡ Denotes functional encryption.
§ Noise accumulation at the aggregation process leads to accuracy drop after a limited number of rounds.

the training process. This class encompasses all ML algorithms.

(2) Private Centralized Learning. The whole dataset is still held by one party, but DP is used as a privacy mechanism in the training process. Works such as [20] can be classified within this category.

(3) None-private Federated Learning. In this scenario, the dataset is distributed to multiple parties, and no privacy mechanisms are applied. Works such as [1] can be categorized within this class.

(4) Local Differential Privacy (LDP). While the dataset is distributed to multiple parties, DP is applied to preserve the privacy of each data holder solely. This approach requires a large number of data parties to achieve satisfactory functionality. In contrast, our approach can achieve the same results with fewer data parties, thanks to our noise reduction strategy. Therefore, our approach can be utilized in both cross-silo and cross-device federated learning settings. Works such as [3] can be categorized within this class.

(5) Hybrid Approach. In this approach, the amount of added noise is reduced using homomorphic encryption to improve accuracy. Works such as [15] can be categorized within this class.

(6) LFR and ERR. Loss Function based Rejection and Error Rate based Rejection are used to defend against poisoning attacks in a non-private federated fashion. Works such as [4] can be categorized within this class.

Comparison in the non-Byzantine setting. Figure 2a shows the testing accuracy results for various approaches mentioned earlier with 20 participants in non-Byzantine background coordinating 50 rounds of training with the privacy parameter $\sigma = 8$. We note that the Hybrid approach and ARPS acquire the same results in a non-Byzantine context; therefore, we consider one of them for representation convenience. As ARPS falls in LDP category, our approach is able to achieve the accuracy of 0.923, which is lower than the best result of 0.989 related to Federated No Privacy, in line with the noisy nature of differential privacy. However, ARPS remarkably outperforms the LDP with the best accuracy score of 0.685, thanks to the noise reduction strategy. As shown in Figure 2a, the smooth oscillations in the performance of the LDP stem from updates overwhelmed by noise.

Comparison in the Byzantine setting. In Figure 2b, and Figure 2c, we demonstrate the convergence performance of our approach in contrast with Hybrid, ERR, and LFR under label-flipping and Gaussian attacks with 20 participants, including 30% Byzantine ones. The privacy parameter σ is set to 8 for ARPS and Hybrid. Figure 2b reveals that ERR and Hybrid collapse completely under label-flipping attack while the accuracy of LFR oscillates sharply within a wide range of 0.147 through 0.839. As plotted in Figure 2c, all the approaches, except for ours, have failed entirely to accomplish the training process under Gaussian attack. However, our approach can reach an accuracy of 0.933, relying on its strong Byzantine robustness characteristic.

Performance evaluation on privacy budget. In Fig-

ure 2d, we pick diverse privacy budgets $\epsilon = 0.6$, $\epsilon = 1.3$, and $\epsilon = 6.4$ to show the results of testing accuracy with 20 participants, including 30% Byzantine participants conducting 50 training iterations. Further, we include a non-private approach with no Byzantine participants to compare with ARPS. As we expected through theoretical results, the accuracy value increases as we relax the privacy protection, which is in inverse relation to ϵ. Meanwhile, our defense provides a convergence guarantee under poisoning attacks.

Remark 3. Our experiments have been conducted within a cross-silo architecture, where FL holds a relatively small number of users who own a large amount of data and are configured with sufficient computational resources. Moreover, we've adopted a local differential privacy model in which the aggregator is considered untrusted, so DP is applied at the user side.

5.2 System Feature Comparison

In Table 1, we summarize state-of-the-art FL approaches to compare their features through the lens of security and efficiency. To address MIA and PA, a functional federated learning framework should observe strong privacy guarantees, Byzantine robustness, and communication efficiency. However, most existing studies either overlook privacy, leading users to lose their trust in the system or lack a mechanism to counteract poisoning attacks, resulting in a denial of service. Only some studies, like [11], claim to provide privacy guarantees and Byzantine robustness, but they leave global models unprotected. Our secure approach provides a frame for learning of high precise ML models without sacrificing efficiency, which is the key contribution of our approach. Table 2 shows the global accuracy (GA), false positive rate (FP), and detection rate (DR) of various attacks on CNN classifier for MNIST, when different defense mechanisms are adopted. DR shows a fraction of honest users who have been correctly identified. Our approach can increase DR by 2% and decrease FP by 3-5% compared to Cosine and Krum in isolation under Gaussian attack. Further, it achieves 100% DR under label-flipping attack. As can be seen, relaxing the privacy loss (ϵ) can improve the GA.

6 Conclusion

In this paper, we take a principled approach in designing an innovative FL scheme to train an ML model in a distributed manner with refined accuracy. We have focused on the most pressing confidentiality and availability concerns in FL, including MIA and PA posing serious threats to data owners and model performance. To overcome information extraction beyond learning models, we draw on DP. By deploying **Sanitize**, an outlier detection strategy adapted for cipher space, we can recognize users who aim to deny the availability of the learned model. On the other hand, our empirical results declare that ARPS can gain high accuracy of 0.93 facing 30% Byzantine participants. Additionally, the communication overhead for every participant is at most $O(1)$ in each iteration. However, our work is confined to untargeted poisoning attacks. It would be an interesting future direction to probe the targeted poisoning attacks, such as backdoor attacks.

Table 2. Testing results of attacks on the CNN classifier for MNIST

Defense	ϵ	#users	Attack	GA(%)	FP(%)	DR(%)
-	∞	10	-	98.76	-	-
Krum	∞	10	Gaussian	98.47	16.66	92.85
Cosine	∞	10	Gaussian	98.55	18.66	92.00
Sanitize	∞	10	Gaussian	98.49	13.6	94.14
Sanitize	∞	10	Label-flipping	98.45	0	100
Sanitize	100	10	Gaussian	92.53	13.9	95.00
Sanitize	15	10	Gaussian	91.31	11.99	94.85

Acknowledgment

The authors would like to thank Information Security and Systems Lab (ISSL) members, and the anonymous referees for their helpful comments on the earlier versions of this article.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[2] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society, 2017.

[3] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[4] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th*

USENIX security symposium (USENIX Security 20), pages 1605–1622, 2020.

[5] Jinhyun So, Başak Güler, and A Salman Avestimehr. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications*, 39(7):2168–2181, 2020.

[6] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[7] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

[8] Yinbin Miao, Ziteng Liu, Hongwei Li, Kim-Kwang Raymond Choo, and Robert H Deng. Privacy-preserving byzantine-robust federated learning via blockchain systems. *IEEE Transactions on Information Forensics and Security*, 17:2848–2861, 2022.

[9] Sana Awan, Bo Luo, and Fengjun Li. Contra: Defending against poisoning attacks in federated learning. In *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*, pages 455–475. Springer, 2021.

[10] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863. IEEE, 2021.

[11] Xu Ma, Yuqing Zhou, Laihua Wang, and Meixia Miao. Privacy-preserving byzantine-robust federated learning. *Computer Standards & Interfaces*, 80:103561, 2022.

[12] Yuan-Cheng Lai, Jheng-Yan Lin, Ying-Dar Lin, Ren-Hung Hwang, Po-Chin Lin, Hsiao-Kuang Wu, and Chung-Kuan Chen. Two-phase defense against poisoning attacks on federated learning-based intrusion detection. *Computers & Security*, 129:103205, 2023.

[13] Ashneet Khandpur Singh, Alberto Blanco-Justicia, and Josep Domingo-Ferrer. Fair detection of poisoning attacks in federated learning on non-iid data. *Data Mining and Knowledge Discovery*, pages 1–26, 2023.

[14] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[15] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11, 2019.

[16] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. Hybridalpha: An efficient approach for privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 13–23, 2019.

[17] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[18] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[19] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

[20] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[21] Jinhyun So, Başak Güler, and A Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *IEEE Journal on Selected Areas in Information Theory*, 2(1):479–489, 2021.

Mojtaba Shirinjani received his B.Sc. degree in Electrical Engineering from the Isfahan University of Technology, Isfahan, Iran, in 2020. Since then, he has been with the Sharif University of Technology, Tehran, Iran, where he is currently pursuing an M.Sc. degree in Electrical Engineering/Secure Communications and Cryptography. His main areas of research interest cover Cryptography, Machine Learning, Neural Networks, Cellular Networks, IoT, Distributed Systems Privacy, and

Blockchain.

Siavash Ahmadi received the B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2012, 2014, and 2020, respectively. He is currently an assistant professor at the Electronics Research Center at the Sharif University of Technology. His special fields of interest include Cryptology, Wireless, Cellular Networks, Internet of Things, and Artificial Intelligence, with an emphasis on security.

Taraneh Eghlidos received her B.Sc. degree in Mathematics from the University of Shahid Beheshti, Tehran, Iran, in 1986, and the M.Sc. degree in Industrial Mathematics from the University of Kaiserslautern, Germany, in 1991. She received her Ph.D. degree in Mathematics from the University of Giessen, Germany, in 2000. She joined the Sharif University of Technology (SUT) in 2002 as a faculty member and is currently an Associate Professor with the Electronics Research Institute at SUT. Her research interests include interdisciplinary research areas, such as Symmetric and Asymmetric Cryptography, Applications of Coding Theory in Cryptography, and Mathematical Modeling for representing and solving real-world problems. Her current fields of research include Lattice-based and Code-based Cryptography.

Mohammad R. Aref received the B.Sc. degree from the University of Tehran, Iran, in 1975, and the M.Sc. and Ph.D. degrees from Stanford University, Stanford, CA, USA, in 1976 and 1980, respectively, all in Electrical Engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. From 1982 to 1995, he was a Faculty Member with the Isfahan University of Technology. He has been a professor of Electrical Engineering at the Sharif University of Technology, Tehran, since 1995 and a distinguished professor since 2013. He has published more than 400 technical articles in Communication and Information Theory, and Cryptography in international journals and conference proceedings. His current research interests include areas of Communication Theory, Information Theory, Cryptography, and Statistical Signal Processing.