Zero-Knowledge with Robust Learning: Mitigating Backdoor Attacks in Federated Learning for Enhanced Security and Privacy

Linlin Li[1], Chungen Xu[2(✉)], and Pan Zhang[1]

[1] School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[2] School of Mathematics and Statistics, Nanjing University of Science and Technology, Nanjing 210094, China
xuchung@njust.edu.cn

Abstract. As a distributed machine learning framework, federated learning addresses the challenges of data isolation and privacy concerns, ensuring that user data remains private during the model training process. However, the privacy-preserving nature of federated learning also makes it has vulnerability to security attacks, particularly in the form of backdoor attacks. These attacks aim to compromise the integrity of the model by embedding a malicious behavior that can be triggered under specific conditions. In our study, aiming to counteract backdoor threats in federated learning, we introduce a new protective mechanism termed zero-knowledge with robust learning (ZKRL). The ZKRL scheme introduces the robust learning rate and non-interactive zero-knowledge proof techniques to filter out malicious model updates and preserve the privacy of the global model parameters of the federated learning process. The extensive experiments conduct on real-world data demonstrate its effectiveness in improving the accuracy on the verification set by 2% and significantly reducing the accuracy of backdoor attacks compared to existing state-of-the-art defense schemes. In summary, the proposed ZKRL defense scheme provides a robust solution for protecting federated learning models against backdoor attacks, ensuring the integrity of the trained models while preserving user privacy.

Keywords: Federated learning · Backdoor attack · Zero-knowledge proof

1 Introduction

Federated learning(FL) [16] is a distributed machine learning paradigm where multiple clients, such as mobile edge devices or corporate organizations, collaborate to train a shared global model under the coordination of a central server. The

Supported by The National Natural Science Foundation of China (No. 62072240).

J. Liu et al. (Eds.): TridentCom 2023, LNICST 523, pp. 99–116, 2024.
https://doi.org/10.1007/978-3-031-51399-2_6

emergence of FL addresses the challenges posed by data dispersion and privacy concerns in traditional centralized machine learning. FL has gained significant interest in both academia and industry. For example, a variety of world-leading companies have introduced various FL application frameworks, such as Google's TensorFlow Federated (TFF) [6] and Webank's FATE [14].

In FL, the privacy of user data is preserved as it remains stored locally on client devices. Instead of sharing raw data, clients only upload encrypted model updates to the central server during the training process. This decentralized approach offers several benefits, including improved model performance compared to traditional local training and full utilization of computing resources of multiple devices, which speeds up model training, etc. However, the nature of FL, where model updates are hidden, also brings a significant security concern: backdoor attacks [3,5]. Backdoor attacks aim to manipulate the global model's behavior by injecting specific inputs (e.g., images with backdoor triggers) that are misclassified into a targeted false label while appearing normal for regular inputs [10,15,21]. Since the server receives only the aggregated results without the prior knowledge of how the local model updates are generated, it becomes challenging to detect whether the local model parameters contain malicious backdoor modifications through direct parameter analysis. Consequently, FL is more susceptible to backdoor attacks, and conventional detection methods prove ineffective in this scenario.

To mitigate the risks of backdoor attacks in FL, researchers have been actively investigating and developing a variety of defense mechanisms and detection techniques specifically tailored for this setting. Several defense methods have been proposed to enhance the security of FL system, including integrity verification, dimension filtering, and robust aggregation algorithms [1,2,24,25]. However, some existing defense methods have certain limitations. The lightweight defense proposed by [17] that adjusts the learning rate may introduce exposure of sensitive user information as it requires processing model updates in plaintext. Additionally, reversing the learning rate for a specific dimension, regardless of the update's origin, can potentially lead to a decrease in the accuracy of the global model. EIFFel introduced by [19] is the verifiably updated security aggregation algorithm. This method involves joint verification by the remaining clients under the supervision of the central server. While it enhances the security of FL by ensuring the integrity of the model updates, it comes at the cost of significant communication overhead and computational burdens. The need for joint verification among clients can result in increased communication and computational costs, which may limit the scalability and efficiency of the FL system.

In this paper, our main task is to address the challenge of backdoor attacks targeting deep neural networks in FL. To achieve this, we propose a scheme named zero-knowledge with robust learning (ZKRL), which filters out malicious model updates during the training process. In ZKRL, we introduce a more refined mechanism for handling model updates at each dimension. Instead of simply multiplying a certain dimension of the model update by a negative learning rate, with the help of zero-knowledge proof, without exposing user sensitive information,

the server checks whether the value of each model update is within the specified range and then judges whether it is a malicious model update. And the data that requires zero-knowledge proof is a certain dimension of the model update rather than the entire model update or even more complex L_2-norm calculations. To evaluate the effectiveness of our defense mechanism, we conduct a series of empirical experiments. The results of these experiments demonstrate that our proposed approach effectively minimizes the loss of model accuracy while successfully deterring backdoor attacks. Moreover, we also provide empirical evidence that supports the effectiveness of our defense mechanism. In addition to the empirical evaluation, we also provide a theoretical analysis that explains how our defense mechanism improves model accuracy. By providing a solid theoretical foundation, we offer insights into the underlying mechanisms that contribute to the effectiveness of our approach.

The structure of this paper is as follows: Sect. 2 covers foundational topics like FL, zero-knowledge proof, and strategies for both backdoor attacks and their defenses. An overview of the solution is sketched in Sect. 3. Section 4 delves into the methods and algorithms underpinning ZKRL. Section 6 elaborates on the empirical testing of ZKRL. The paper wraps up in Sect. 7 with our conclusions.

2 Preliminaries

2.1 Notation

Let $\mathbb{Z}_p$ denote the ring of integers modulo p, where p is a large prime. $\mathbb{G}_q$ represents the unique subgroup of $\mathbb{Z}_p$ of order q, where q divides $p-1$. Let $\mathbb{G}_q^n$ and $\mathbb{Z}_p^n$ be vector spaces of dimension n over $\mathbb{G}_q$ and $\mathbb{Z}_p$ respectively, and $\alpha = (g_1, \cdots, g_n)$, $\beta = (h_1, \cdots, h_n) \in \mathbb{G}_q^n$ be a vector of generators. For a scalar $c \in \mathbb{Z}_p$ and a vector $a \in \mathbb{Z}_p^n$, we denote by $b = c \cdot a \in \mathbb{Z}_p^n$ the vector where $b_i = c \cdot a_i$. Furthermore, let $\langle a, b \rangle = \sum_{i=1}^{n} a_i \cdot b_i$ denote the inner product between two vectors $a, b \in \mathbb{Z}_p^n$, and $a \bigodot b = (a_1 \cdot b_1, \cdots, a_n \cdot b_n) \in \mathbb{Z}_p^n$ denote the Hadamard product or entry wise multiplication of two vectors.

2.2 Federated Learning

FL was first proposed by Google, which is a distributed machine learning framework that enables a central server to collaboratively train a model with a large number of decentralized clients. The process begins by initializing the global model parameters W_G on the central server. Prior to each round of FL, a subset of m clients (where $m < n$) is randomly chosen from a total of n clients to participate in the training. In the t^{th} round, the central server broadcasts the current global model W_G^t to the selected clients. Each client then performs local training on its own dataset using stochastic gradient descent(SGD). This local training yields a new local model W_i^{t+1} for the i^{th} client. Subsequently, the i^{th} client uploads its model update $\Delta w_i^{t+1} = W_i^{t+1} - W_G^t$ to the central server. The central server employs a secure aggregation algorithm to merge the received

model updates to update the global model. Here, the aggregation rule of the Federated Averaging (FedAvg) algorithm, one of the commonly used aggregation algorithms in FL, is as follows:

$$W_G^{t+1} = W_G^t + \eta \bigodot \frac{1}{m} \sum_{i \in m} \Delta w_i^{t+1}. \tag{1}$$

This iterative process of model distribution, local training, and secure aggregation allows the central server and clients to collaboratively train a robust global model while preserving data privacy and security.

2.3 Zero-Knowledge Proof

Zero-knowledge proof [13], initially proposed by Goldwasser et al., is a cryptographic protocol that involves two parties: the prover P and the verifier V. It is commonly used for proof of membership attribution or proof of knowledge. Zero-knowledge proof possesses three essential properties: completeness, soundness, and zero-knowledge. Completeness guarantees the correctness of the protocol itself. If the prover P provides a valid proof of a statement, and both P and V execute the protocol honestly, then P can convince V of the statement's correctness. Soundness ensures that the honest verifier V cannot be deceived by a malicious prover $P^{'}$. It protects V from being misled or convinced of a false statement. Zero-knowledge property means that the prover P can prove the correctness of a statement to the verifier V without revealing any information other than the statement's validity. This property ensures privacy and confidentiality. The three properties of zero-knowledge proof make it suitable for establishing trust and protecting privacy in FL, especially for checking model updates. However, interactive proofs require P and V to be continuously online, which can be challenging to achieve due to network delays, denial of service attacks, and other factors. Non-interactive proofs, on the other hand, require the prover to send only one round of messages to complete the proof. Interactive proofs can be transformed into non-interactive proofs using the Fiat-Shamir heuristic [11], which helps to overcome the limitations of interactivity.

2.4 Backdoor Attacks and Defense Solutions

Backdoor Attacks in Federated Learning. A backdoor attack involves embedding a hidden trigger in a model that can be activated to produce specific misclassifications without affecting the model's inference on normal data. In this attack, the adversary aims to manipulate the model's behavior to perform targeted misclassifications based on the presence of the trigger. For instance, in the case of a self-driving model, the backdoor may cause the model to correctly recognize a regular stop sign but misclassify a stop sign with a yellow square as a speed limit sign, with the yellow square acting as the trigger. In a unified environment, a malicious actor might manipulate the training to instill targeted misclassification traits for a backdoor assault. But in FL, given its decentralized

data nature, gaining access to the complete training dataset poses challenges for the adversary. That said, backdoor attacks often revolve around formulating harmful modifications. Essentially, the attacker aims to craft a modification embedding the backdoor such that, upon aggregation with other alterations, the merged model manifests the undesired backdoor trait. This assault mode is often termed model poisoning attack [3,5,23]. In such a scenario, the opponent might control several agents involved in the FL process, training their local models using tainted or compromised datasets to produce harmful updates.

Defense Solutions. Several studies have explored the application of robust statistics techniques to enhance the resilience of FL against attacks. In a notable work by [12], Fung et al. proposed a novel approach to improve the robustness of FL by introducing a per-client learning rate, deviating from the conventional practice of employing a uniform learning rate at the server side. This novel approach results in the following update rule:

$$W_G^{t+1} = W_G^t + \frac{1}{m} \sum_{i \in m} \eta_i^t \cdot \Delta w_i^{t+1}, \tag{2}$$

where $\eta_i^t \in [0, 1]$ is the i^{th} client's learning rate for the t^{th} round.

CRFL, proposed by [24], is a protective strategy that employs trimming and refining processes on parameters to ensure the global model's consistency. This method facilitates individual sample resilience assurance against backdoor incursions. In contrast, Ozdayi et al. [17] addressed backdoor attacks by adjusting the learning rate of the aggregation server in a per-dimension and per-round manner, utilizing the sign information of the client's updates.

3 Problem Overview

3.1 Security Goals

Input Privacy (Client's Goal). The first security goal is to safeguard user data privacy in FL. In this decentralized approach, the raw user data remains on the local device, with only model updates being transmitted to the central server. However, it is possible for an honest but curious server to infer information about the original data by analyzing the model updates. To address this concern, clients need to employ cryptographic primitives to guarantee that no party can gain knowledge about the raw input (update) Δw_i of a client C_i, beyond what can be deduced from the final aggregated result W_{agg}.

Input Integrity (Server's Goal). The second security goal is to safeguard the security of the global model. As model updates are concealed, the server lacks the ability to verify the integrity of these updates, consequently raising the risk of potential backdoor attacks initiated by malicious clients. Hence, it becomes imperative for the server to implement measures to filter out any malicious model updates while ensuring the preservation of input privacy, ultimately fortifying the global model against backdoor attacks.

3.2 Threat Model

Semi-honest Server. In this paper, we assume that the server is semi-honest, meaning that it will adhere to the protocol rules but may attempt to infer secret information by observing its own perspective during the execution process. Furthermore, we also take into account that the server possesses a portion of the dateset on its local device for training purposes.

Malicious Clients. We assume the presence of e malicious clients, who have the ability to take arbitrary actions during the protocol execution. However, for the purpose of conducting a backdoor attack in FL, these e malicious clients can only train a malicious model locally and subsequently upload a malicious model update to introduce a backdoor into the global model.

3.3 Solution Overview

Previous research has made attempts to address the issue of backdoor attacks in FL by implementing robust aggregation algorithms or verifiable algorithms in the ciphertext state. However, these methods have not fully met our security objectives. Taking into account our analysis of the threat model, we propose a novel FL scheme called ZKRL, which effectively filters out malicious model updates. To ensure input privacy, we employ Pedersen commitments [18], and we enhance input integrity by integrating non-interactive Bulletproof range proofs [8] into the robust learning rate (RLR) technique proposed by [17]. The ZKRL scheme selectively aggregates model updates that conform to predefined criteria of well-formedness, thereby preserving the integrity of the global model and protecting it against potential backdoor attacks. The overview of our proposed ZKRL scheme is shown in Fig. 1.

4 ZKRL Design

4.1 Secure Aggregation with Commitments

To ensure the privacy of the model updates in FL, it is essential to employ effective encryption techniques. In our proposed approach, we utilize the Pedersen commitment [18] as our cryptographic primitive of choice. The Pedersen commitment provides a secure and efficient approach of encrypting the model updates while preserving the privacy of the underlying data. The Pedersen commitment possesses two important properties: unconditionally hiding and computationally binding. The unconditionally hiding property ensures that the committed value remains hidden and cannot be determined by any party, even if they possess unlimited computational power. On the other hand, the computationally binding property guarantees that once a commitment is made, it is computationally infeasible to change the committed value without detection. One of the notable advantages of using Pedersen commitments is their homomorphic property. This property allows for the computation of operations on the commitments

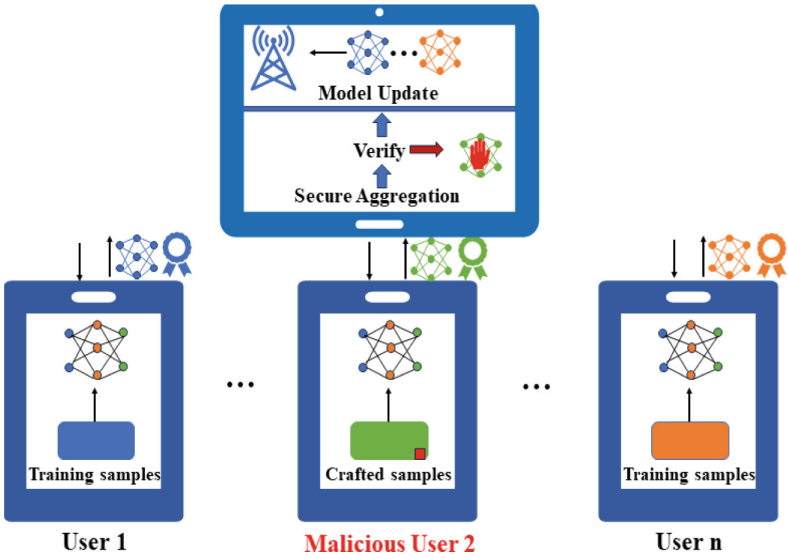

Fig. 1. Overview of ZKRL.

themselves, such as addition and subtraction, without the need to decrypt the underlying values. This enables efficient and secure aggregation of the encrypted model updates without compromising the privacy of the individual updates. Furthermore, Pedersen commitments facilitate the construction of zero-knowledge range proofs. These proofs enable verification of the committed values' validity within a specified range without revealing any information about the actual values. This property ensures the integrity and correctness of the encrypted model updates without disclosing any sensitive information.

To achieve unconditional hiding, a random number $s \in \mathbb{Z}_p$ is selected as the blinding factor. We select a generator g and a random group element $h \in \mathbb{G}_q$, such that nobody knows $\log_g h$. In the protocol, the i^{th} client commits to each element u_j of the model update Δw_i^{t+1}, and subsequently sends the resulting commitment c_j, which is calculated as follows:

$$c_j = g^{u_j} h^{s_j}, \tag{3}$$

where s_j represents the blinding factor specific to that client. To ensure accurate aggregation of model updates, we adopt the masking approach [7] using blinding factors in the commitment-based setting. This approach allows for the cancellation of blinding factors during aggregation, enabling the server to obtain the aggregated output accurately. In our protocol, each of m participating clients selects a blinding factor, denoted as s_i, such that the sum of all the blinding factors is equal to zero: $\sum_{i=1}^{m} s_i = 0$. This ensures that blinding factors cancel out during the aggregation phase. Given two Pedersen commitments c_1 and c_2 for values u_1 and u_2 respectively, the product of these commitments, $c_1 \cdot c_2$,

corresponds to a commitment for the sum of u_1 and u_2 $(u_1 + u_2)$. By applying this property to each dimension of the model updates, we can aggregate the commitments efficiently. Specifically, the aggregation is computed as follows:

$$\prod_{i=1}^{m} c_{j,i} = g^{\sum_{i=1}^{m} u_{j,i}} h^{\sum_{i=1}^{m} s_{j,i}} = g^{\sum_{i=1}^{m} u_{j,i}}. \tag{4}$$

Finally, the server can derive the sum of all the model updates by calculating the discrete logarithm to the base g. While calculating the discrete logarithm of $y = g^x$ is generally a computationally hard problem, in our domain, the space of the aggregated result is small, allowing for efficient computation of the discrete logarithm [20,22]. It is important to note that in the FL scenario, we often need to encrypt negative numbers. To handle this, we encode negative numbers before encryption. For a negative number A, we compute $A^{'} = A \bmod p$, where p is set to twice the value range of the model updates. This ensures that negative numbers do not overlap with positive numbers after encoding. Algorithm 1 formally describes the process for secure aggregation with commitments.

Algorithm 1: Secure Aggregation with Commitments.

Input: $u_{j,i}$ is the j^{th} dimension of the model update Δw_i^{t+1} of the i^{th} client C_i, l is the dimension of the model updates, generator g $\in \mathbb{G}_q$, random group element h $\in \mathbb{G}_q$.

Output: The aggregated model update Δw_G^{t+1}.

Client C_i jointly decides the choice of blinding factor $s_{j,i}$;
for i=1 $\rightarrow$ m **do**
  for j=1 $\rightarrow$ l **do**
    C_i computes the commitment $c_{j,i} = g^{u_{j,i}} h^{s_{j,i}}$, and sends $c_{j,i}$ to the central server;
for j=1 $\rightarrow$ l **do**
  The server S computes $agg_j = g^{\sum_{i=1}^{m} u_{j,i}} h^{\sum_{i=1}^{m} s_{j,i}} = g^{\sum_{i=1}^{m} u_{j,i}}$, and $u_{j,G} = \log_g agg_j$;
$\Delta w_G^{t+1} = (u_{1,G}, u_{2,G}, \cdots, u_{l,G})$;
Return Δw_G^{t+1}.

4.2 Zero-Knowledge Range Proof

Bulletproofs [8] is a zero-knowledge proof protocol proposed by Bootle et al. that offers several advantages, including short proofs and the absence of a trusted setup phase. It is notably adept for proficient range proofs on committed values. It enables the confirmation of a value's presence within a range with just $2 \log n +$ 9 group and field elements, with n being the bit length of the range. Additionally, Bulletproofs natively supports the Pedersen commitments. In our approach, we

leverage the Bulletproofs protocol to enable the server to verify model update uploaded by clients. Specifically, in certain cases (as elaborated in the subsequent analysis section), the server may request clients to provide a range proof for a specific dimension updated by its own model, ensuring that the value falls within a specified range. To transform this proof process into a non-interactive one, we apply the Fiat-Shamir heuristic [11].

Formally, let $u_{j,i}$ be a number in $[0, \cdots, 2^n - 1]$, and $c_{j,i}$ be a Pedersen commitment to $u_{j,i}$ using randomness $s_{j,i}$, $2^n - 1$ is the specific range that the server requires the client to prove. The proof system uses the homomorphic property of the Vector Pedersen commitments to construct commitments to two polynomials $l(X)$ and $r(X)$ in $\mathbb{Z}_p^n[X]$. With these vector-polynomial pledges, the server and clients participate in an inner product dialogue to reliably determine the inner product between $l(X)$ and $r(X)$. The structure of these polynomials ensures that the zero coefficient of $\langle l(X), r(X)\rangle \in \mathbb{Z}_p[X]$ takes a distinct shape solely when $u_{j,i}$ is within the specified range. Let $a_L = (a_1, \cdots, a_n)$ be the vector containing the bits of $u_{j,i}$, the client C_i commits to a_L as well as blinding vectors s_L, s_R using constant sized vector commitments, and constructs the polynomial $t(X) \in \mathbb{Z}_p[X]$ as a function of $u_{j,i}$, $t(X)$ is exactly the inner product of $l(X)$ and $r(X)$:

$$t(X) = \langle l(X), r(X)\rangle \in \mathbb{Z}_p[X]. \tag{5}$$

Here, s_L, s_R whose zero coefficient is independent of $u_{j,i}$ if and only if $u_{j,i}$ indeed contains only bits. The client executes Algorithm 2 to generate a proof of its own model update.

The server receives the zero-knowledge range proof sent by the clients and executes Algorithm 3 for verification.

4.3 ZKRL Workflow

Ozdayi et al. proposed a method where the malicious model update aims to minimize the loss function of both the main task and the backdoor attack task, while the honest model update only focuses on minimizing the loss function of the main task. As a result, these two types of model updates may have different or even opposite directions in certain dimensions. Building upon this observation, the RLR was introduced to involve multiply the learning rate by a negative value to maximize the loss function of the backdoor attack task. However, this method has its limitations. While it successfully maximizes the loss function of the backdoor attack task, it may also impact dimensions related to honest model updates, resulting in a decrease in the accuracy of the global model. Additionally, the method still is performed in plaintext, which does not align with our objective of ensuring input privacy.

Therefore, we propose ZKRL, a novel approach based on the aforementioned method. When the sum of signs of the model updates at a certain dimension j is less than the learning threshold θ, we employ a different treatment instead of simply inverting the learning rate. The server analyzes its own local model

Algorithm 2: Zero-Knowledge Range Proof.

Input: $2^n - 1$ is the specific range that the server requires the clients to prove, $c_{j,i} = g^{u_{j,i}} h^{s_{j,i}}$ is the commitment corresponding to a certain dimension of the model update that needs to give a range proof, H is a hash function, $g, h \in \mathbb{G}_q$, $\alpha, \beta \in \mathbb{G}_q^n$.

Output: zero-knowledge range proof RP.

$a_L \in \{0,1\}^n \; s.t. \; \langle a_L, 2^n \rangle = u_{j,i}$;
$a_R = a_L - 1^n$;
$\gamma \xleftarrow{\$} \mathbb{Z}_p$;
$A = h^{\gamma} \alpha^{a_L} \beta^{a_R}$;
$s_L, s_R \xleftarrow{\$} \mathbb{Z}_p^n$;
$\rho \xleftarrow{\$} \mathbb{Z}_p$;
$S = h^{\rho} \alpha^{s_L} \beta^{s_R}$;
$y = H(A, S)$;
$z = H(A, S, y)$;
$l(X) = a_L - z \cdot 1^n + s_L \cdot X$;
$r(X) = y^n \circ (a_R + z \cdot 1^n + s_R \cdot X) + z^2 \cdot 2^n$;
$t(X) = \langle l(X), r(X) \rangle = \sum_{d=0}^{2} t_d \cdot X^d$;
$\tau_1, \tau_2 \xleftarrow{\$} \mathbb{Z}_p$;
$T_d = g^{t_d} h^{\tau_d}, d = \{1, 2\}$;
$x = H(T_1, T_2)$;
$\tau_x = \tau_1 \cdot x + \tau_2 \cdot x^2 + z^2 \cdot s_{j,i}$;
$\delta = \gamma + \rho \cdot x$;
$l = l(x), r = r(x), t = \langle l, r \rangle$;
$RP = (A, S, y, z, T_1, T_2, x, \tau_x, \delta, t, l, r)$;
Return RP.

update and determines the value $w_{j,s}$ corresponding to dimension j. Based on the value of $w_{j,s}$, we handle the model updates as follows:

If $w_{j,s}$ is not equal to 0, the server publishes $w_{j,s}$ and requires all clients to provide proof that the value $w_{j,c}$ corresponding to the dimension of their local model update is less than $w_{j,s}$. If $w_{j,s}$ is negative, indicating a malicious model update, the server removes it from the aggregation process. If $w_{j,s}$ is positive, indicating an honest model update, the server proceeds with aggregation.

If $w_{j,s}$ is equal to 0, we apply the method proposed by [17] to directly invert the learning rate.

Compared to the method proposed by [17], our ZKRL scheme retains honest model updates to a greater extent when abnormal sign detection occurs, resulting in higher accuracy of the final global model. For instance, in our proposed scheme, we categorize model updates into two groups based on the value of $w_{j,s}$ and the outcome of the inspection process. This categorization allows us to filter out the malicious updates and apply a normal learning rate for updating the honest models during the aggregation process. In contrast, the method introduced by [17] does not differentiate the quality of the model updates. Instead, it directly

Algorithm 3: Check Zero-Knowledge Range Proof.

Input: $RP = (A, S, y, z, T_1, T_2, x, \tau_x, \delta, t, l, r)$ is the range proof that the client sends to the server, $c_{j,i} = g^{u_{j,i}} h^{s_{j,i}}$ is the commitment corresponding to a certain dimension of the model update that needs to give a range proof, $g, h \in \mathbb{G}_q$, $\alpha, \beta \in \mathbb{G}_q^n$.

Output: True or False.

Server computes $t^{'} = \langle l, r \rangle$;
if $t^{'} = t$ **then**
  Continue;
else
  Return False;
k(y,z) = $-z^2 \cdot \langle 1^n, y^n \rangle - z^3 \cdot \langle 1^n, 2^n \rangle$;
Server computes $g^{k(y,z)+z \cdot \langle 1^n, y^n \rangle} \cdot c_{j,i}^{z^2} \cdot T_1^x \cdot T_2^{x^2}$ and $g^t h^{\tau_x}$;
if $g^{k(y,z)+z \cdot \langle 1^n, y^n \rangle} \cdot c_{j,i}^{z^2} \cdot T_1^x \cdot T_2^{x^2} = g^t h^{\tau_x}$ **then**
  Continue;
else
  Return False;
for $d{=}1 \rightarrow n$ **do**
  $h_d^{'} = h_d^{y^{-d+1}}$;
Server computes $P = AS^x \cdot \alpha^{-z \cdot 1^n} \cdot \beta^{'z \cdot y^n + z^2 \cdot 2^n}$;
if $P = h^{\delta} \alpha^l h^{'r}$ **then**
  Continue;
else
  Return False;
Return True.

employs the opposite learning rate to update the models in all dimensions. By subdividing the model updates based on their quality, our approach enables a more fine-grained evaluation and filtering process, resulting in improved defense against malicious updates. This distinction allows us to selectively update the models with a more appropriate learning rate, thus enhancing the overall robustness and integrity of the FL system. The execution flow of protocol ZKRL is shown in Algorithm 4.

5 Security Analysis

In this section, we formally analyze the security of ZKRL.

Theorem 1. *ZKRL scheme satisfies the security goals of privacy protection and integrity verification at the same time.*

Algorithm 4: ZKRL.

Input: m is the number of clients selected in one iteration, W_G^t is the global model parameters for thr t^{th} iteration, l is the dimension of the model updates, θ is the learning threshold, η is the learning rate.

Output: global model paramaters for the $(t+1)^{th}$ iteration W_G^{t+1}.

Server S distributes the global model W_G^t to clients;
for i=1→m **do**
  Client C_i trains W_G^t using it's data D_i locally to achieve local model W_i^{t+1}, $\Delta w_i^{t+1} = W_i^{t+1} - W_G^t$. C_i commits to Δw_i^{t+1} using Pedersen commitment, and send it to the server along with the sign of Δw_i^{t+1};
Server S also trains W_G^t using its data D_s locally to achieve local model W_s^{t+1}, $\Delta w_s^{t+1} = W_s^{t+1} - W_G^t$;
for j=1→l **do**
  Server S computes $sgn_j = \sum_{i=1}^{m} sgn(u_{j,i})$;
  if $sgn_j \geq \theta$ **then**
    $\eta_{\theta,j} = \eta$;
  else if $sgn_j < \theta$ **then**
    if $\Delta w_{j,s}^{t+1} = 0$ **then**
      $\eta_{\theta,j} = -\eta$;
    else
      Server S publishes the value $\Delta w_{j,s}^{t+1}$ and requires m clients to use Algorithm 2 to generate a proof that $\Delta w_{j,i}^{t+1}$ is less than $\Delta w_{j,s}^{t+1}$;
    for i=1→m **do**
      Client C_i uses Algorithm 2 to generate a proof RP_i, and sends it to the server;
      Server S uses Algorithm 3 to check RP_i and get verification result V_i;
      if V_i = *True and* $\Delta w_{j,s}^{t+1} > 0$ **then**
        $\Delta w_{j,i}^{t+1}$ is honest;
      if V_i = *True and* $\Delta w_{j,s}^{t+1} < 0$ **then**
        $\Delta w_{j,i}^{t+1}$ is malicious;
      if V_i = *False and* $\Delta w_{j,s}^{t+1} > 0$ **then**
        $\Delta w_{j,i}^{t+1}$ is malicious;
      if V_i = *False and* $\Delta w_{j,s}^{t+1} < 0$ **then**
        $\Delta w_{j,i}^{t+1}$ is honest;
Server S filters out malicious model updates and uses Algorithm 1 to obtain aggregated model update Δw_G^{t+1};
$W_G^{t+1} = W_G^t + \eta_\theta \bigodot agg(\Delta w_G^{t+1})$;
Return W_G^{t+1}.

Proof. ZKRL utilizes the Pedersen commitment to ensure the privacy of model updates, this scheme relies on the properties of the Pedersen commitment, where for any element $u \in \mathbb{Z}_p$ and for randomly uniformly chosen $s \in \mathbb{Z}_p$ and $g, h \in \mathbb{G}_q$, $g^u h^s$ is uniformly distributed in $\mathbb{G}_q$. Formally, if $u, u^{'} \in \mathbb{Z}_p$ satisfies $u \neq u^{'}$ and $g^u h^s = g^{u^{'}} h^{s^{'}}$, then it must hold that $s \neq s^{'} \bmod p$ and $\log_g h = \frac{u-u^{'}}{s^{'}-s} \bmod p$. However, it is not feasible to directly calculate the value of $\log_g h$. And ZKRL achieves the integrity verification of the model through the use of range proofs. In ZKRL, range proofs are utilized due to their beneficial attributes, such as perfect completeness, perfect honest verifier zero-knowledge, and computational special soundness.

6 Experimental Evaluation

In this section, we conduct a comprehensive performance analysis of our proposed defense mechanism and provide empirical evidence to support its effectiveness. To evaluate the performance, we implemented a FL prototype system using the PyTorch framework. The hardware setup includes an AMD Ryzen 7 (2.90 GHz) CPU, an RTX2080 (8 GB) GPU, and 64 GB of memory. The software environment consists of the Ubuntu 20 operating system, Python 3.7, and PyTorch 1.13 with Cudn 10.1.

In our experiments, we created a FL scenario with a total of 100 participants, including both honest participants and malicious participants. The original dataset was divided into 100 subsets, with each participant receiving one subset. It is important to note that the distribution of samples across different categories was kept uniform, ensuring that all participants had independently and identically distributed (IID) data. Furthermore, we also carried out tests with the Federated EMNIST dataset sourced from the LEAF benchmark [9], specifically examining the performance under the Non-IID setting.

6.1 Experimental Setup

Datasets. To evaluate the efficiency of our introduced ZKRL method within FL, various classification exercises are designed using these three datasets:

MNIST Dataset: This collection comprises 70,000 hand-drawn numeric images spanning 10 categories (0–9), with 60,000 designated for training and 10,000 for testing. To ensure consistency, we normalized all samples in the MNIST dataset to a standardized size of 28×28 pixels.

CIFAR-10 Dataset: The CIFAR-10 collection encompasses 60,000 colored pictures across 10 categories (like airplanes, vehicles, birds, and so on), partitioned into 50,000 training instances and 10,000 for testing. During data preparation, pictures from the CIFAR-10 set are adjusted to a 32×32 three-layer format.

Federated EMNIST Dataset: The Federated EMNIST dataset is a variant of the Extended MNIST (EMNIST) dataset specifically designed for federated learning scenarios. EMNIST is an extension of the MNIST dataset, which consists of handwritten digits from 0 to 9. This dataset distribution simulates the decentralized nature of federated learning, where each participant holds their own local data without sharing it directly with the central server or other participants.

Model Settings. In this experiment, we employ the same model architecture as described in [23]. The model is a 5-layer convolutional neural network with approximately 1.2 million parameters. It consists of two convolutional layers, followed by a max-pooling layer, and then two fully-connected layers with dropout. This architecture has been previously demonstrated to be effective in various tasks and serves as a baseline for our experiments.

Evaluation Metrics. We evaluate ZKRL's performance using three criteria: validation precision, primary class precision, and backdoor precision. The first two, validation and primary class precision, are calculated based on the datasets' validation data used in our tests. Backdoor precision, on the other hand, is determined using an altered validation set, where all entries from the primary class undergo a trojan transformation and are re-categorized as the target class. Our analysis juxtaposes ZKRL's outcomes with those of four different aggregation approaches: FedAvg [16], FedAvg with RLR [17], FoolsGold [12], and sign aggregation [4]. These methods are widely used in FL research and provide a basis for comparison with our proposed ZKRL scheme.

6.2 ZKRL Defense Performance Analysis

IID Setting. We performed tests with the MNIST and CIFAR-10 datasets, distributing the data evenly among agents following an i.i.d. approach. We equally allocated training samples to every agent via consistent sampling. Figure 2 showcases the learning trajectories of three techniques: FedAvg, FedAvg incorporated with RLR, and FedAvg paired with ZKRL. Sequentially from left to right, the trajectories highlight validation precision, primary class precision, and backdoor precision. As can be seen, it is evident that the FedAvg method alone is vulnerable to the backdoor attack. However, when using the RLR and ZKRL defense mechanisms, the backdoor attack is effectively prevented. Moreover, in comparison to RLR, ZKRL not only successfully mitigates the backdoor attack but also leads to improvements in accuracy rates on both the training and validation sets. The base class accuracy shows an increase of 1.2% to 1.5%, indicating better performance on the original classes, and the validation accuracy demonstrates a notable improvement of 0.8%. Additionally, Table 1 summarizes the final accuracy achieved by these methods. The lowest backdoor precision and the highest validation and primary class precision are emphasized in bold. The results clearly demonstrate the effectiveness of our proposed ZKRL scheme in providing significant protection against backdoor attacks compared to the baselines.

Table 1. Final validation, primary class and backdoor precision for different aggregations in i.i.d. setting.

Aggregation	Validation (%)	Base (%)	Backdoor (%)
FedAvg (MNIST)	93.2	**99.1**	100
FedAvg (CIFAR-10)	**93.4**	98.9	100
FoolsGold (MNIST)	93.3	98.5	100
FoolsGold (CIFAR-10)	93.1	98.6	100
Sign (MNIST)	93.1	98.6	99.7
Sign (CIFAR-10)	92.9	98.7	99.8
FedAvg with RLR (MNIST)	92.2	97.4	0.5
FedAvg with RLR (CIFAR-10)	92.0	97.3	0.1
FedAvg with ZKRL (MNIST)	93.0	98.9	0.4
FedAvg with ZKRL (CIFAR-10)	92.8	98.5	**0**

These findings validate the superiority of our proposed approach in mitigating the impact of backdoor attacks, thereby enhancing the overall performance and security of FL systems.

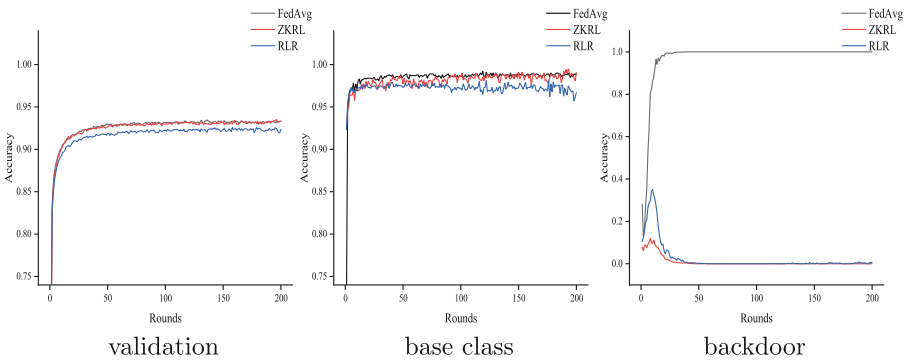

Fig. 2. Training curves for FedAvg, FedAvg with the ZKRL and FedAvg with the RLR on MNIST dataset.

Non-IID Setting. We shift our focus to a scenario in FL that mirrors a more practical data distribution: non-i.i.d. For this evaluation, we employ the Federated EMNIST dataset sourced from the LEAF benchmark. In a manner analogous to the i.i.d. setup, we study the learning trajectories of FedAvg, FedAvg paired with ZKRL, and FedAvg combined with RLR, as depicted in Fig. 3. The final accuracy outcomes for each aggregation methods are summarized in Table 2.

The lowest backdoor precision and the highest validation and primary class precision are emphasized in bold. The experimental results demonstrate that ZKRL is effective in resisting backdoor attacks even in the non-i.i.d. environment. Furthermore, ZKRL achieves improved accuracy rates compared to RLR. Specifically, the validation accuracy improved by 2% and the base accuracy increased by 0.9%. These findings highlight the robustness and effectiveness of ZKRL in the presence of data heterogeneity among agents.

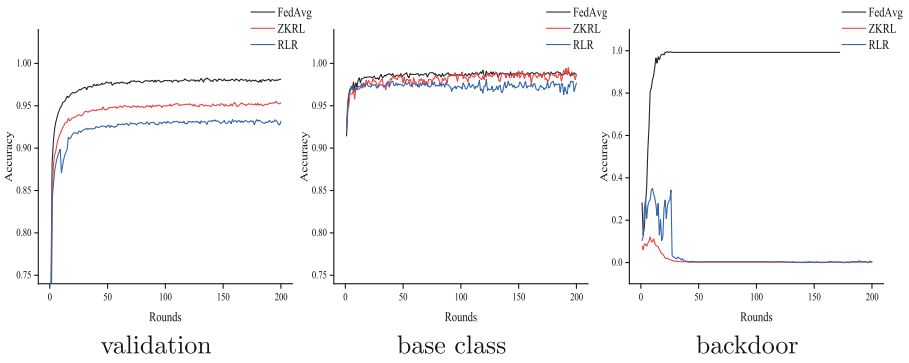

Fig. 3. Training curves for FedAvg, FedAvg with the ZKRL and FedAvg with the RLR on Federated EMNIST dataset.

Table 2. Final validation, primary class and backdoor precision for different aggregations in non-i.i.d. setting.

Aggregation	Validation (%)	Base (%)	Backdoor (%)
FedAvg	98.0	**98.7**	**99.2**
FoolsGold	97.9	98.6	99.1
Sign	97.8	98.5	99.7
FedAvg with RLR	93.2	97.7	0.4
FedAvg with ZKRL	95.2	98.6	**0.2**

7 Conclusion

In this work, we conducted a comprehensive investigation into FL with a specific emphasis on countering adversarial attacks, particularly backdoor attacks. Our defense mechanism is centered around analyzing the relationship between the model update and the value announced by the server, enabling us to dynamically adjust the fine-tuning learning rate. This adjustment is performed on a

per-dimension and per-round basis, leveraging the sign information of clients' updates and the outcome of the inspection process. Through extensive experimentation, we demonstrated the effectiveness of our defense mechanism in significantly reducing backdoor accuracy while maintaining minimal impact on the overall validation accuracy. In fact, our defense outperformed several state-of-the-art defense techniques reported in the existing literature. As part of our future work, we plan to explore avenues for accelerating the cryptographic primitives utilized in our defense mechanism by leveraging GPU hardware. By enhancing the efficiency of our defense, we aim to make it more applicable and adaptable to real-world scenarios.

References

1. Andreina, S., Marson, G.A., Möllering, H., Karame, G.: BaFFLe: backdoor detection via feedback-based federated learning. In: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), pp. 852–863. IEEE (2021)
2. Awan, S., Luo, B., Li, F.: CONTRA: defending against poisoning attacks in federated learning. In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021, Part I 26. LNCS, vol. 12972, pp. 455–475. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88418-5_22
3. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics, pp. 2938–2948. PMLR (2020)
4. Bernstein, J., Zhao, J., Azizzadenesheli, K., Anandkumar, A.: signSGD with majority vote is communication efficient and fault tolerant. arXiv preprint arXiv:1810.05291 (2018)
5. Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S.: Analyzing federated learning through an adversarial lens. In: International Conference on Machine Learning, pp. 634–643. PMLR (2019)
6. Bonawitz, K., et al.: Towards federated learning at scale: system design. Proc. Mach. Learn. Syst. **1**, 374–388 (2019)
7. Bonawitz, K., et al.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191 (2017)
8. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 315–334. IEEE (2018)
9. Caldas, S., et al.: LEAF: a benchmark for federated settings. arXiv preprint arXiv:1812.01097 (2018)
10. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
11. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
12. Fung, C., Yoon, C.J., Beschastnikh, I.: Mitigating sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866 (2018)
13. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 203–225 (2019)

14. Liu, Y., Fan, T., Chen, T., Xu, Q., Yang, Q.: FATE: an industrial grade platform for collaborative learning with data protection. J. Mach. Learn. Res. **22**(1), 10320–10325 (2021)
15. Liu, Y., et al.: Trojaning attack on neural networks. In: 25th Annual Network And Distributed System Security Symposium, NDSS 2018. Internet Society (2018)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
17. Ozdayi, M.S., Kantarcioglu, M., Gel, Y.R.: Defending against backdoors in federated learning with robust learning rate. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9268–9276 (2021)
18. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
19. Roy Chowdhury, A., Guo, C., Jha, S., van der Maaten, L.: EIFFeL: ensuring integrity for federated learning. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 2535–2549 (2022)
20. Shafagh, H., Hithnawi, A., Burkhalter, L., Fischli, P., Duquennoy, S.: Secure sharing of partially homomorphic encrypted IoT data. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, pp. 1–14 (2017)
21. Shafahi, A., et al.: Poison frogs! Targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
22. Shi, E., Chan, H., Rieffel, E., Chow, R., Song, D.: Privacy-preserving aggregation of time-series data. In: Annual Network & Distributed System Security Symposium (NDSS). Internet Society (2011)
23. Sun, Z., Kairouz, P., Suresh, A.T., McMahan, H.B.: Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963 (2019)
24. Xie, C., Chen, M., Chen, P.Y., Li, B.: CRFL: certifiably robust federated learning against backdoor attacks. In: International Conference on Machine Learning, pp. 11372–11382. PMLR (2021)
25. Zhang, J., Ge, C., Hu, F., Chen, B.: RobustFL: robust federated learning against poisoning attacks in industrial IoT systems. IEEE Trans. Industr. Inf. **18**(9), 6388–6397 (2021)