For Thirty Pieces of Silver: Overcoming Non-IID via ZKP Sample-Level Auditing and Forcing Attackers into Traitors in Federated Learning

Jinghao Zhao, Hongwei Yang, Bobo Wang, Meng Hao, Hui He, and Weizhe Zhang*

***Abstract*—As a privacy-preserving distributed learning paradigm, federated learning faces severe security threats, especially data poisoning, gradient tampering, distributed attacks, and adaptive attacks. Existing gradient-based detection methods encounter three major bottlenecks in practical applications. First, they cannot distinguish between benign and malicious gradients under distributed attacks in non-IID scenarios. Second, defenders lack prior knowledge of trigger patterns. Third, attackers can launch adaptive attacks against these defense mechanisms. To address these challenges, we propose a novel defense paradigm. This paradigm shifts away from gradient detection, instead leveraging zero-knowledge proofs to conduct sample-level inspections on the local datasets of clients, eliminating the defense degradation caused by distributed attacks under non-IID settings. Then, through a staking mechanism, attackers are forced to train on discovered malicious samples with flipped labels, generating counter-gradients to neutralize hidden malicious samples, achieving targeted backdoor removal without requiring any prior knowledge of backdoor features. Furthermore, through zero-knowledge proof commitments, training gradients must be generated based on the same locked data. This prevents dynamic adaptive attacks based on reference model parameters and precludes gradient tampering. Experimental results demonstrate that under the highly non-IID setting of Dir(0.1), our proposed scheme successfully defends against distributed backdoor attacks and edge-case attacks on CIFAR-10, suppressing the attack success rate to below 6%. To the best of our knowledge, this is the first framework to achieve such defense results.**

***Index Terms*—Federated Learning, Zero-Knowledge Proofs, Data Poisoning, Verifiable Computing.**

I. Introduction

Federated learning, as an emerging distributed learning paradigm, allows mutually untrusted clients to collaboratively train models while keeping their datasets locally private [1]–[4]. However, due to this data opacity, federated learning faces severe security threats [5], [6]. Attackers can easily launch poisoning attacks, primarily including data poisoning (such as implanting small triggers or modifying predicted labels) [7], [8] and gradient tampering (directly manipulating uploaded model gradients or weights) [9]–[12]. In addition, attackers may launch adaptive attacks, dynamically adjusting attack strategies during training based on parameters broadcast by the defender [11]–[15]. Attackers can also control multiple nodes to launch distributed attacks (also known as Sybil attacks) [9], [16]–[19].

Although research on poisoning detection in federated learning has made some progress, most existing methods rely on the heuristic analysis of gradients [18], [20]–[26], facing three major bottlenecks in practical applications. First, in non-IID data scenarios, gradients from honest clients inherently exhibit significant variance. Consequently, gradient-based detection algorithms easily confuse benign updates with malicious poisoning, allowing attackers to disperse malicious samples across different clients to evade detection. Second, benign clients only possess knowledge of benign sample features, whereas malicious clients know both benign and malicious features, creating a severe capability asymmetry between attackers and defenders. Third, the lack of a strong cryptographic binding between the training data and the computation process allows clients to tamper with data or directly forge computed gradients, leaving enormous room for adaptive attacks.

To address these challenges, we propose a novel federated learning defense paradigm that provides a systematic solution to the three aforementioned bottlenecks.

To overcome the challenges of non-IID data and distributed attacks, we recognize that traditional gradient inspection is inherently unreliable under such harsh conditions. Therefore, rather than relying on gradient analysis, we leverage zero-knowledge proof (ZKP) technology to perform privacy-preserving, sample-level inspections directly on the clients' local datasets. Because the global data distribution does not affect the loss evaluation of individual samples, this approach eliminates the impact of non-IID data and distributed attacks during the detection stage.

Furthermore, to address the capability asymmetry between benign and malicious clients, we argue that discovered malicious samples are, in a sense, more valuable than benign ones because they contain complete trigger information. Thus, instead of discarding these attack samples, we employ ZKPs and a deposit-staking mechanism to force attackers to correct the labels of their suspicious samples. This compels them to use their own trigger information to produce gradient updates that counter their original attack intent. By incorporating these betrayed malicious samples into the defender's camp, we effectively reverse the capability asymmetry between the defender and attackers.

Finally, to counter adaptive and gradient tampering attacks, we introduce a cryptographic data-binding mechanism. The original data is first locked via the underlying state of the

Jinghao Zhao, Hongwei Yang, Bobo Wang, Meng Hao, Hui He, and Weizhe Zhang are with Harbin Institute of Technology, Harbin, China (e-mail: zhaojinghao@stu.hit.edu.cn; yanghongwei@hit.edu.cn; wangbobochn@hit.edu.cn; haomeng@hit.edu.cn; hehui@hit.edu.cn; wzzhang@hit.edu.cn).

*Corresponding author: Weizhe Zhang (e-mail: wzzhang@hit.edu.cn).

ZKP protocol before the detection algorithm parameters are distributed. ZKPs are then used to guarantee that the training gradients are generated from this exact, unaltered data. This precludes gradient forgery and prevents dynamic adaptive attacks based on reference model parameters.

Specifically, this defense paradigm consists of three stages. In the initialization stage, clients commit to their local data through the setup phase of the ZKP protocol and pay a staking deposit upon registration. Concurrently, the server pre-trains a reference detection model using a small clean dataset. In the ZKP audit stage, clients locally compute the loss value of each sample using the reference model and submit ZKPs to verify computation authenticity and data consistency. The server then marks samples with anomalous loss values in each class as suspicious. At this point, the defender neither discards these suspicious samples nor penalizes the clients; instead, it simply relabels them to match the reference model's predictions. In the secure federated training stage, when clients participate in formal gradient updates, they must continuously prove the consistency between their training data and the initially committed state. Any forged proof, or any training refusal by clients with suspicious samples, will result in the full forfeiture of their staking deposit.

Extensive experiments demonstrate that under typical non-IID settings, our method significantly outperforms existing baselines in defending against data poisoning attacks. By conducting sample-level inspections based on loss values, our paradigm successfully identifies a large number of suspicious samples prior to model training. Additionally, our approach prevents gradient tampering and dynamic adaptive attacks targeting reference model parameters. Notably, even under the highly non-IID setting of $\mathrm{Dir}(0.1)$ [27], we successfully defend against the Distributed Backdoor Attack (DBA) [17] and the Edge-Case Backdoor Attack (ECBA) [15] on the CIFAR-10 dataset.

In summary, our main contributions are threefold:

- We propose a novel defense paradigm based on sample-level inspection, which eliminates the impact of non-IID data distributions and distributed attacks on the detection of malicious samples.
- We design a "betrayal" mechanism that repurposes detected poisoned samples as defensive counter-samples, forcing attackers to utilize their own triggers against their original backdoors. This effectively resolves the capability asymmetry between benign and malicious clients.
- We introduce a cryptographic state-binding mechanism that protects clients' private data while enforcing strict temporal consistency between data and behavior. This eliminates gradient tampering attacks and effectively prevents dynamic adaptive attacks against the reference model.

II. Background

A. Zero-Knowledge Proofs

In our defense framework, the ZKP protocol is treated as an orthogonal cryptographic primitive. To satisfy our security requirements, the verifiable computation backend must strictly guarantee the following four properties:

- **Completeness:** If the prover honestly computes a public function $y = f(x)$ using a valid private input x, the verifier must always accept the generated proof π. Formally:
$$\Pr[\mathrm{Verify}(f, y, \pi) = 1 \mid y = f(x)] = 1$$
- **Soundness:** A malicious prover cannot fabricate a valid proof π' for an incorrect computation. The probability of deceiving the verifier when $y \neq f(x)$ is bounded by a negligible function ϵ:
$$\Pr[\mathrm{Verify}(f, y, \pi') = 1 \mid y \neq f(x)] \leq \epsilon$$
- **Zero-Knowledge:** The proof π must not leak any information about the private input x. A polynomial-time simulator Sim must be able to generate a view computationally indistinguishable from the real execution, using only public variables:
$$\mathrm{View}_V^{\mathcal{P}(x)}(f, y) \approx_c \mathsf{Sim}(f, y)$$
- **Commitment and Cross-State Binding:** The prover must bind the private state x to a succinct, unforgeable public commitment $C = \mathrm{Commit}(x, r)$, where r denotes a cryptographic blinding factor. Furthermore, the protocol must enforce global consistency by cryptographically linking temporally separated independent circuits to this exact same initial commitment C, mathematically preventing state equivocation.

B. Threat Model

The federated learning literature currently encompasses a vast array of attack variants, many of which share similar logic, differ only in specific configurations (e.g., trigger patterns), or combine multiple underlying mechanisms. To provide a clear threat model without enumerating redundant methods, we abstract these complex adversaries into four fundamental, composable categories based strictly on their physical execution mechanisms.

We consider a powerful active adversary $\mathcal{A}$ attempting to compromise the global aggregated model to fit its malicious intent, while simultaneously evading system detection to prevent the forfeiture of its deposited collateral. $\mathcal{A}$ possesses the full spectrum of local control, enabling the following attack vectors:

- **Gradient Tampering:** During the protocol execution, $\mathcal{A}$ deviates from honest local training and directly modifies model updates. Typical methods in this category include Sign-flipping [9], extreme Gaussian noise injection [20], and Model Replacement via malicious gradient scaling [14]. The shared characteristic of these attacks is bypassing the raw dataset entirely to manually alter the magnitude or direction of the submitted gradients in the parameter space.
- **Data Poisoning:** $\mathcal{A}$ has complete control over local datasets and can arbitrarily poison the training source. This encompasses the most prolific class of standard attacks, including random label-flipping for untargeted

degradation, as well as targeted backdoor attacks such as BadNets [8], blended watermark injection [7], and localized pixel patches. The fundamental mechanism always lies in altering the genuine mapping between data features and ground-truth labels prior to the training process.

- **Adaptive Attacks:** $\mathcal{A}$ constructs updates to maximize the attack impact while evading the detection thresholds of the deployed defenses. Prominent strategies include A Little Is Enough [12], Min-Max [11], and Fall of Empires [13]. The adversary rigorously constrains the updates to stay within the statistical variance of benign gradients, thereby remaining undetected while covertly biasing the global model.
- **Distributed Poisoning:** $\mathcal{A}$ possesses sufficient financial resources to deposit margins and control massive malicious nodes. This enables advanced collusion strategies, prominently including DBA [17] and Sybil-based targeted attacks [9]. Instead of submitting a single, conspicuously anomalous update, the adversary either decomposes malicious features into subtle fragments across different nodes [17] or coordinates Sybil updates to jointly bias the aggregation [9], thereby diluting individual attack signatures to bypass isolated defenses.

C. Defense Model

In this mechanism, the central aggregator acts as the defender $\mathcal{S}$, whose primary objective is to maintain the integrity of the global model and penalize malicious nodes. We assume $\mathcal{S}$ possesses a small, clean private dataset used exclusively to pre-train a reference detection model θ_{ref}.

The defender $\mathcal{S}$ holds the following authorities and capabilities:

- **Label Correction:** The defender audits the loss of all client samples evaluated on θ_{ref}. For samples exhibiting anomalous loss, $\mathcal{S}$ mandates the clients to correct their labels to the predictions output by θ_{ref}.
- **Deposit Forfeiture:** The defender has the authority to confiscate the entire deposit of clients committing the following violations: (1) forging ZKPs; (2) refusing to continue participating in the training after their sample labels have been forcibly corrected.

III. Related Work

A. Traditional Defense Methods

Traditional defense mechanisms against poisoning and backdoor attacks usually do not rely on cryptographic tools; instead, they directly analyze gradients uploaded by users. These methods adopt various strategies, such as distance-based clustering, historical similarity tracking, and the use of auxiliary validation data.

Among early proposals, foundational methods primarily depend on strict statistical boundaries or global similarity assumptions. For instance, Krum [20] selects a single local model that minimizes the sum of squared distances to its nearest neighbors. However, this winner-takes-all mechanism discards massive amounts of benign information, leading to significant model degradation under non-IID data where no single client captures the global distribution. Similarly, Auror [23] applies k-means clustering to discard deviating parameters, but struggles against multi-backdoor threats. To counter Sybil attacks, FoolsGold [19] exploits historical cosine similarity; yet, it theoretically fails to detect the inherently weak similarity of fragmented updates in DBA. Other approaches incorporate auxiliary mechanisms: BaFFLe [28] relies on local client validation, which depends on the often circumvented assumption that backdoors inevitably degrade clean-task accuracy, while clients cannot effectively isolate malicious degradation from severe local data skew. Alternatively, FLTrust [21] relies on a clean server-side dataset, but its gradient similarity metrics inherently misclassify highly skewed benign updates under extreme non-IID settings.

Recent works have evolved into more sophisticated pipelines constructed to be robust against varying initial data distributions. FLAME [24] utilizes HDBSCAN dynamic clustering alongside adaptive clipping and noise injection. FLDetector [29] predicts client updates via L-BFGS to flag significant consistency discrepancies without requiring server-side validation data. Furthermore, CrowdGuard [30] integrates Trusted Execution Environments (TEE) with layer-wise output metrics (HLBIM) using PCA. Although the aforementioned methods have reported promising results under IID or mildly non-IID settings, their theoretical foundations fail when faced with extreme non-IID scenarios. Under such extreme settings, criteria such as update consistency or density clustering inadvertently capture a large amount of benign variance in data heterogeneity rather than malicious patterns. In addition, dimensionality reduction techniques are inherently subject to physical limitations; in the early stage of the training process, random initialization noise dominates the parameter space, disrupting PCA metrics.

In summary, conventional defenses are fundamentally constrained by their reliance on gradient statistics. Under extreme non-IID conditions, the natural variance of benign data heavily overlaps with malicious signatures. This overlap effectively neutralizes traditional distance and density anomaly detection. Furthermore, early-stage initialization noise frequently disrupts statistical tools like dimensionality reduction. These limitations demonstrate that dynamic gradient filtering alone is insufficient. Consequently, robust security requires alternative evaluation paradigms that operate independently of gradient distribution statistics.

B. Zero-Knowledge Proof-Based Defenses

Recent research has introduced ZKPs to federated learning to address the privacy-utility trade-offs of traditional defenses. ZKP-based frameworks allow clients to prove the correctness of their local computations without revealing the raw data. The primary objective of these approaches is to enforce computational integrity and ensure compliance with predefined security rules, thereby protecting the training process from malicious manipulation while maintaining data privacy.

Some methods utilized ZKPs to securely enforce traditional gradient inspection rules on the server side. For instance, CustodianFL [31] guarantees the execution of a 3σ distance-based filter, while RankFL [32] secures the sorting and pruning

of encrypted gradients. Because these methods rely on global gradient-space statistics, they degrade under severe non-IID conditions: benign minority-class updates are disproportionately pruned as outliers, whereas attackers can scale poisoned gradients to bypass the statistical median bounds.

To mitigate server-side privacy risks and aggregation bottlenecks, subsequent frameworks shift the inspection mechanism to the client side. RoFL [33] integrates Bulletproofs with homomorphic commitments to enforce L_2 and L_∞ norm bounds on encrypted client updates. While effective against single-shot model replacement, RoFL's own analysis reveals that norm bounds cannot prevent continuous backdoor attacks targeting tail distributions or using artificial triggers, as such attacks embed backdoors without requiring large gradient magnitudes. Following this paradigm, Gupta et al. [34] employ zk-SNARKs to ensure that locally computed gradient histograms fall within globally preset bounds. However, this reliance on static global bounding introduces a difficult trade-off under non-IID settings: accommodating natural data variance requires wide bounds that permit backdoor injection, while strict bounds reject benign, highly-skewed clients whose local gradients naturally deviate. Similarly, Nguyen and Thai [35] require clients to execute localized backdoor defense algorithms (e.g., pruning dormant neurons) and prove compliance via ZKPs. This local inspection logic inherently assumes IID data distributions; in severe non-IID scenarios, valid neurons responsible for missing classes on benign clients remain unactivated and are thus falsely penalized by the pruning algorithm, leading to the incorrect rejection of honest updates.

To address the limitations of continuous gradient inspection, Zhang et al. [36] map updates into discrete Top-k sparse sign votes, utilizing ZKPs to bound the voting magnitude. While this discrete voting scheme preserves main task accuracy against untargeted attacks, its effectiveness against targeted backdoor attacks remains limited. Because the optimization on rare benign data and the injection of backdoor triggers produce similar coordinate shifts in the discrete space, attackers can still embed backdoors while complying with the local ZKP magnitude constraints.

However, existing ZKP-based defenses primarily guarantee the computational correctness or magnitude compliance of the training process rather than the benign nature of the underlying data. Because these methods rely on gradient norm bounds [33], [34] or local pruning rules [35], they remain vulnerable to stealthy data poisoning in non-IID scenarios. Attackers can craft poisoned samples, such as tail-distribution or distributed backdoor triggers, that strictly comply with the required cryptographic constraints and bypass verification. This indicates that enforcing update-level constraints alone is insufficient to prevent data-level poisoning. Consequently, there remains a critical need for defense strategies that directly inspect and bind sample-level semantics while maintaining the privacy guarantees of federated learning.

IV. Design

A. Framework Overview

Our approach consists of the following steps:

1) The clients commit to their private datasets through the initialization phase of the ZKP protocol and deposit a security margin to the platform. The server trains a reference detection model using its small private dataset.
2) The server publishes the parameters of the reference model and requires clients to compute and submit the loss values of their private data on this model. Clients must generate ZKPs to verify the authenticity of the loss computation and the consistency of the data with the initial state established in the first step.
3) Based on the collected loss distribution, the server identifies and flags samples with abnormally high loss values, and requires the clients holding these suspicious samples to train them using the labels predicted by the reference model.
4) During the formal federated training phase, clients must provide ZKPs to ensure that the input data used to generate gradients is identical to the data bound during initialization.
5) If forged ZKPs are detected during the process, or if a client possessing suspicious samples refuses to participate in the training, the system will immediately confiscate that client's entire deposit.
6) Upon the successful completion of the federated training, clients who have not triggered any penalty conditions will receive a full refund of their deposits. Optionally, the system may distribute training rewards to these honest clients in proportion to their initial deposits.

The technical implementations of Step 1, Steps 2–3, and Steps 4–6 in our framework are elaborated in Section IV-C, Section IV-D, and Section IV-E, respectively.

B. Role of ZKP in Defense

In our defense framework, ZKP is the most critical infrastructure. To ensure the integrity of the loss-based poisoning detection mechanism and the subsequent local training process without compromising client data privacy, the ZKP protocol enforces three key checks within the framework:

1) Preventing loss forgery during the detection phase: The core of our defense relies on identifying poisoned samples through abnormally high loss values under the reference model. The ZKP protocol forces clients to honestly compute the loss of each sample on the reference model, preventing attackers from attempting to bypass the flagging mechanism by submitting forged loss values.

2) Preventing cross-stage data replacement: Attackers may use benign data during the detection phase to evade flagging and replace it with poisoned data during the training phase. To this end, the protocol requires clients to pre-generate a cryptographic commitment to their local data. The subsequent loss computation and gradient generation proofs are forced to be anchored to this same commitment, mathematically eliminating data swapping while preserving privacy.

3) Preventing gradient forgery in local training: To prevent attackers from directly manipulating or forging gradients, the system enforces verification that the submitted gradients are derived strictly from the global model parameters and

the aforementioned data locked by the commitment, thereby eliminating any possibility of attacks based on modifying gradients.

It is worth noting that the specific underlying implementation of zkML is not the focus of this paper. We regard it as orthogonal work that can be implemented by existing standard zkML frameworks [37]–[42]. In the architectural design of this paper, we use these zero-knowledge operators as black boxes.

C. Phase 1: Cryptographic Data Binding and Initialization

Building upon the ZKP foundation, the cryptographic binding of client datasets is natively integrated into the setup phase (Step 1). Let the private dataset of client i be $\mathcal{D}_i = \{(x_{i,j}, y_{i,j})\}_{j=1}^{m_i}$, consisting of m_i private feature-label pairs. Treating $\mathcal{D}_i$ as the underlying private state, the client generates a succinct cryptographic commitment $C_i = \mathrm{Commit}(\mathcal{D}_i, r_i)$, and submits C_i to the server. This commitment globally binds client i to its exact dataset $\mathcal{D}_i$ across all subsequent protocol phases without leaking the plaintext.

The chronological order of the process prevents adaptive attacks against the reference model parameters. Clients must lock their data states via C_i before the server publishes the reference model parameters, denoted as θ_{ref}. If adversaries could observe θ_{ref} prior to generating C_i, they could adaptively tailor their poisoned samples to yield normal loss values $\mathcal{L}(x, y; \theta_{\mathrm{ref}})$ for $(x, y) \in \mathcal{D}_i$ and evade detection. By enforcing state initialization in Step 1, attackers are forced to commit $\mathcal{D}_i$ blindly. Due to the binding property of the commitment, any ex post facto alteration of $\mathcal{D}_i$ to fit θ_{ref} is computationally infeasible and will fail subsequent ZKP verification.

During the initialization phase, client i is also required to pay a substantial security deposit to participate in the protocol. This deposit serves to deter malicious clients from forging proofs associated with C_i or refusing to cooperate in the subsequent computational processes.

D. Phase 2: Loss Detection and Flagging

Unlike traditional federated poisoning detection, which relies on continuously monitoring gradients during the training phase, our inspection mechanism (Steps 2 and 3) only needs to be executed once during the initialization phase before formal training. This avoids the global model already being contaminated when anomalies are detected. More importantly, it transforms client-level inspection into sample-level inspection. Below, we introduce the advantages of sample-level inspection and how to dynamically adjust category thresholds.

1) Advantages of Sample-level Inspection: This mechanism provides three major advantages:

First, regarding immunity to non-IID data interference, this method is immune to the interference caused by non-IID data. In federated learning, the essence of non-IID is the imbalanced data partition at the client level (i.e., substantial discrepancies exist in the distribution $P_i(x, y)$ across local datasets $\mathcal{D}_i$). Traditional gradient inspection methods fail under non-IID scenarios because the local gradient $g_i = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathcal{D}_i} \nabla\mathcal{L}(x, y; \theta)$ is an expectation calculated over the entire local dataset $\mathcal{D}_i$. This implies that the gradient metric is strongly coupled with the client's data distribution. When a client's data is highly skewed, the gradient deviation induced by objective data skewness highly overlaps in statistical features with poisoned gradients manipulated towards malicious directions, leading to the failure of defense mechanisms.

In contrast, our defense mechanism employs a single-sample loss evaluation $\mathcal{L}(x, y; \theta_{\mathrm{ref}})$. This computation strictly occurs between a fixed reference model θ_{ref} and an independent data sample (x, y). This sample-level evaluation mathematically decouples from the client-level data partition. In other words, the loss value of a normal sample (x, y) computed under the reference model depends solely on the mapping relationship between its own features and label, and has absolutely no correlation with which client it is allocated to, or whether that client's overall label distribution is extremely skewed or uniform. Because the calculation dimension of the detection metric (sample-level) directly bypasses the dimension where non-IID occurs (client-level), distribution skewness at the client level cannot impose any interference on the loss value of the sample itself.

Second, concerning the distributed stealth strategies, this shift in inspection granularity from the "client-level" to the "sample-level" eliminates the dilution-based stealth strategies of distributed attacks. In traditional gradient monitoring, attackers often dismantle triggers and distribute them across multiple clients, utilizing local normal samples to dilute malicious gradient deviations and hide them within massive normal statistical noise. However, under the single-sample loss evaluation mechanism, detection is no longer shielded by the physical dispersion of clients. Independent loss evaluation against θ_{ref} strips away the statistical cover of benign local data, directly exposing tampered samples regardless of their physical dispersion. Therefore, sample-level loss flagging eliminates the dilution effect in distributed attacks.

Third, in terms of eliminating collateral data loss, this granularity shift resolves the collateral data loss intrinsically tied to client-level filtering. Discarding a compromised client inevitably drops all its co-located benign samples. Given a malicious client fraction γ and an intra-client poisoning ratio ρ, an idealized 0% client-level False Positive Rate mathematically equates to a sample-level FPR of $\frac{\gamma(1-\rho)}{1-\gamma\rho}$. In stealthy attacks where $\rho \to 0$, this equivalent data loss approaches γ. By independently evaluating each instance, our sample-level inspection strictly isolates poisoned data and safely retains the uncorrupted samples within malicious nodes, thereby maximizing the preservation of the global data distribution.

2) Dynamic Adjustment of Category Thresholds: A naive global loss threshold overlooks the natural heterogeneity of loss distributions across different classes. Normal samples from harder classes may be more likely to exceed the global threshold and thus be mistakenly flagged. Moreover, attackers can exploit the reference model's insufficient recognition of harder classes by choosing to poison within these classes.

To eliminate the interference of this inherent class imbalance and accurately locate high-loss anomalies caused by poisoned data, we introduce a class-based adaptive threshold adjustment mechanism (see Algorithm 1). To prevent a large number of normal low-loss samples from diluting the statistical deviation,

Algorithm 1 Class-based Adaptive Threshold Flagging

Require: $\mathcal{T}$: Verified sample loss tuples $\{(j, c_j, l_j)\}$, α: Penalty coefficient, β: Base threshold multiplier, C: Set of classes
Ensure: $\mathcal{I}_{\text{flag}}$: Set of flagged sample indices
▷ Step 1: Compute global statistics across all verified losses
$L \leftarrow \{l_j \mid (j, c_j, l_j) \in \mathcal{T}\}$
$L_{\text{global}} \leftarrow \text{Mean}(L)$
$H_{\text{global}} \leftarrow \{l \in L \mid l > L_{\text{global}}\}$
$\tilde{L}_{\text{global}} \leftarrow \text{Mean}(H_{\text{global}})$
$\mathcal{I}_{\text{flag}} \leftarrow \emptyset$
▷ Step 2: Compute class-specific thresholds and flag
for all $c \in C$ **do**
  $L_c \leftarrow \{l_j \mid (j, c_j, l_j) \in \mathcal{T},\, c_j = c\}$
  $\bar{L}_c \leftarrow \text{Mean}(L_c)$
  ▷ Extract high-loss subset for class c
  $H_c \leftarrow \{l \in L_c \mid l > \bar{L}_c\}$
  $\tilde{L}_c \leftarrow \text{Mean}(H_c)$
  ▷ Calculate adaptive flagging threshold
  $\tau_c \leftarrow L_{\text{global}} \cdot \beta \cdot \exp\left(-\alpha\left(\frac{\tilde{L}_c}{\tilde{L}_{\text{global}}} - 1\right)\right)$
  ▷ Identify sample indices exceeding threshold
  for all $(j, c_j, l_j) \in \mathcal{T}$ such that $c_j = c$ **do**
    if $l_j > \tau_c$ **then**
      $\mathcal{I}_{\text{flag}} \leftarrow \mathcal{I}_{\text{flag}} \cup \{j\}$
    end if
  end for
end for
return $\mathcal{I}_{\text{flag}}$

this mechanism specifically isolates the high-loss subset to compute the penalty ratio, thereby significantly amplifying detection sensitivity.

The core of this formula lies in dynamically adjusting the threshold through an exponential penalty term $e^{-\alpha(\frac{\tilde{L}_c}{\tilde{L}_{\text{global}}}-1)}$: when the high-loss average $\tilde{L}_c$ of a certain class c is significantly higher than the global high-loss average $\tilde{L}_{\text{global}}$, the system automatically lowers the flagging threshold for that class. There are two main reasons why the average loss of a specific class is abnormally high: first, that class is the attack target and is mixed with many poisoning samples that exhibit severe feature-label mismatch; second, the reference model largely lacks classification ability for that class, making it difficult to identify the attacker's attack, and its likelihood of being attacked is higher. In both cases, tightening the flagging threshold for that class is a necessary defense strategy.

E. Phase 3: Subverting Malicious Samples

The detection phase eliminates the interference of non-IID data and distributed attacks, while ZKP commitments deter adaptive evasion. However, because the reference model is trained on a small dataset, it identifies only a subset of malicious samples. Furthermore, the defender lacks prior knowledge of the trigger patterns. To address this, we repurpose the flagged samples as a defensive weapon (a strategy we term the "betrayal of malicious samples").

Specifically, the protocol requires clients to relabel the flagged samples using the reference model's predictions and generate new ZKP commitments for these labels. This cryptographic binding ensures the attacker applies the modifications. Contrary to intuition, we do not punish nodes holding flagged samples (e.g., by prohibiting participation or deducting deposits). This strategy not only protects honest nodes, but also demonstrates our "sincerity" in co-opting malicious ones. The attacker faces two choices: (1) be a "traitor", train on both the label-flipped and the undetected samples, where conflicting gradient updates cancel the trigger effect; or (2) abort training, resulting in deposit confiscation.

Just as most betrayals in history stem from the temptation of money, to analyze how much benefit is needed to make an attacker be a traitor, we establish the following game-theoretic model.

After the detection phase, the attacker must choose for each controlled client: cooperate with the relabeling and continue federated training, or disconnect to cut losses. We define the game parameters as follows: Let N be the total global poisoning samples across M malicious clients, yielding $n = \frac{N}{M}$ malicious samples per node. Let $p \in [0, 1]$ denote the expected sample-level detection rate. Because our audit relies on cryptographically state-bound raw data rather than uploaded gradients, this expectation p is strictly independent of M or non-IID distribution. Furthermore, let $R_p > 0$ be the positive payoff of an undiscovered sample penetrating the global model, $L_c > 0$ the negative payoff of a detected and server-flipped sample, and $D > 0$ the initialization staking deposit.

For a specific client, the number of detected malicious samples k follows $k \sim \text{Binomial}(n, p)$, leaving $n - k$ latent samples. The attacker evaluates two strategies:

- Disconnect (or forge ZKP proofs): The attacker forces the client offline, avoiding gradient uploads. The protocol dictates full deposit confiscation, yielding a payoff of $u_{\text{disconnect}}(k) = -D$.
- Cooperate: The client completes the ZKP and uploads gradients containing the flipped samples, retrieving deposit D. The $n - k$ latent samples implant the trigger, yielding $(n-k)R_p$, while the k flipped samples neutralize the trigger, inflicting $-kL_c$. The payoff is $u_{\text{cooperate}}(k) = (n - k)R_p - kL_c$.

A rational attacker maximizes the single-node expected payoff: $u_{\text{opt}}(k) = \max(-D,\, (n-k)R_p - kL_c)$. The attacker abandons the deposit only when cooperation is strictly less profitable than disconnection. Solving $(n-k)R_p - kL_c < -D$ yields the disconnection threshold:

$$k > \frac{nR_p + D}{R_p + L_c}$$

Setting the minimum integer threshold $k^* = \lfloor \frac{nR_p+D}{R_p+L_c} \rfloor + 1$, the global expected total payoff for the attacker controlling M nodes is:

$$U_{\text{Total}} = M \cdot \mathbb{E}_{k \sim \text{Binomial}(n,p)} \left[\max\Big(-D,\ \ (n-k)R_p - kL_c \Big) \right]$$

TABLE I: Ablation Analysis of Defense Modules under Various Attack Vectors

Comb.	M1: Sample Inspection	M2: ZKP Binding	M3: Flipping & Staking	Main Task Accuracy	Data Poisoning	Gradient Tampering	Adaptive Attack	Distributed Poisoning
1	×	×	×	×	×	×	×	×
2	✓	×	×	×	×	×	×	×
3	×	✓	×	✓	×	✓	✓	×
4	×	×	✓	×	×	×	×	×
5	✓	✓	×	✓ or ×	× or ✓	✓	✓	✓
6	✓	×	✓	×	×	×	×	×
7	×	✓	✓	✓ or ×	× or ✓	✓	✓	✓
8	✓	✓	✓	✓	✓	✓	✓	✓

(Note: ✓ indicates defensive capability / high MTA; × indicates defense failure / low MTA; "✓ or ×" and "× or ✓" denote a mutually constraining trade-off where both cannot be achieved simultaneously.)

Expanding this expectation based on the threshold k^* yields:

$$U_{\text{Total}} = M\left(\sum_{k=0}^{k^*-1}\binom{n}{k}p^k(1-p)^{n-k}\big[(n-k)R_p - kL_c\big] + \sum_{k=k^*}^{n}\binom{n}{k}p^k(1-p)^{n-k}\big[-D\big]\right)$$

This game-theoretic formulation highlights three core advantages of the proposed defense. First, a rational attacker will not blindly increase M to launch a distributed attack, which not only significantly increases the attacker's deposit amount, but also does not change the detection probability p. Second, increasing the deposit D significantly raises the disconnection threshold k^*, making the attacker more inclined to become a traitor. Finally, even if they choose to become a traitor, the existence of L_c will also prevent them from obtaining any benefit. Through this mechanism, we effectively turn the tables on the adversary, transforming the defender's uncertainty regarding unknown trigger patterns into the attacker's uncertainty when confronting an unknown reference model. This makes rational attackers abandon poisoning from the outset.

F. Ablation Analysis of the Defense Architecture

To evaluate the effectiveness and interdependencies of the proposed defense components, we conduct a theoretical ablation analysis. We decompose the defense architecture into three core modules: loss-based sample-level inspection (M1), ZKP-state-based cryptographic data binding (M2), and label flipping with an anti-escape staking mechanism (M3). Table I summarizes the theoretical defense capabilities of these modules under different combinations.

Based on the analysis of the table above, we draw the following key conclusions regarding the system design:

1. ZKP binding is the foundation of the defense system (Combinations 1, 2, 4, 6)

Any combination lacking M2 (ZKP binding) fails against all attack vectors. Without a foundational cryptographic mechanism to enforce the locking of the initial data state, the system cannot verify data consistency across different stages of the protocol. Attackers can submit benign data during the detection phase to bypass M1's audit, and subsequently execute a data replacement attack during the actual federated training phase to introduce malicious samples with triggers. M2 blocks the operational space for such evasion tactics and effectively defends against gradient tampering and dynamic adaptive attacks.

2. Limitations of relying solely on ZKP (Combination 3)

Combination 3 represents a defense paradigm relying exclusively on computational integrity verification. This represents some works that attempt to introduce ZKP into federated learning [42], and these works use ZKP to ensure computational correctness. Although ZKP (M2) can mathematically prove with strictness that the gradients are computed from the committed input data, it cannot evaluate the legitimacy of the data at the semantic level. If attackers commit to malicious samples with triggers at the initial stage, the ZKP mechanism can only prove that the computational process of the malicious gradient is correct. This indicates that computational integrity alone is insufficient to guarantee model security; M1 and M3 must be introduced for intervention at the data semantic level.

3. The dilemma between poisoning defense and model performance (Combinations 5 and 7)

These two combinations reveal the inevitable security and performance trade-off the system faces when lacking a complete closed-loop defense:

Combination 5: With the introduction of M1, the benefits of distributed attacks attempting to bypass detection during the aggregation phase by dispersing triggers are eliminated, as sample-level inspection renders any distributed camouflage meaningless. However, without M3, the system can only directly discard suspected high-loss samples. This places the defender in a threshold dilemma: to prevent a small number of low-loss malicious samples from evading detection and corrupting the model, the defender must set a strict filtering threshold. This inevitably leads to the erroneous deletion of a large number of hard-to-learn normal samples (high false positive rate), thereby causing a significant drop in MTA. Conversely, if the threshold is relaxed to maintain MTA, the data poisoning defense will fail.

Combination 7: In the absence of M1 for precise localization, forcefully activating the label modification of M3 leads to blind strategy execution. If labels are modified on a large scale

to cover potential poisoned samples, the data distribution of a vast number of normal samples will be destroyed, leading to a substantial decrease in MTA. If only small-scale modifications are made to maintain MTA, the latent poisoned samples cannot be effectively counteracted. However, since M3 still operates at the sample level, distribution does not give the attacker any gain.

4. Resolving the security-performance dilemma via label flipping (Combination 8)

Combination 8 (M1+M2+M3) effectively resolves the aforementioned dilemma, achieving a balance between model security and performance. M1 can identify the majority of suspected high-loss samples, while the flipping mechanism of M3 alters the traditional defense logic: the system no longer demands a 100% recall rate for malicious samples, and it simultaneously avoids directly discarding data.

By enforcing label flipping on the detected high-loss malicious samples, the system preserves the original data features to maintain a high MTA. Simultaneously, by utilizing the opposite gradients generated by these flipped samples during training, it directly offsets the negative impact of the few undetected low-loss poisoned samples during the model aggregation phase.

Furthermore, combined with the staking mechanism of M3, if malicious nodes discover that their poisoned samples have been converted into defensive data by the system, they will face the financial loss of stake forfeiture if they choose to exit midway. If they continue to participate, they must consume their own computational resources to offset their original attack intentions. This mechanism reshapes the attack and defense costs from the underlying architecture, effectively eliminating the asymmetric offense-defense capabilities in traditional federated learning.

V. Experiments

A. Experimental Setup

The experiments use the standard CIFAR-10 dataset [43], partitioned across clients following a Dirichlet distribution $\mathrm{Dir}(0.1)$. Both the global network and the local client networks adopt the ResNet-18 architecture [44]. The federated learning setup consists of 100 clients and runs for a total of 1500 rounds. In each round, 25 clients are randomly sampled to perform 3 local epochs. The local batch size is set to 64. The optimizer is stochastic gradient descent, with an initial learning rate of 0.01 and a momentum parameter of 0.9. The proportion of malicious clients is set to 30%, the local poisoning rate is 20%, and the target class is specified as class 5 regardless of the samples' source labels. The server reference model is trained using 500 data samples. We assume that the attacker is rational enough not to launch any attacks that are easily detected (e.g., forging ZKPs, actively disconnecting, or forging gradients), but will deploy adaptive attacks alongside advanced backdoor strategies, specifically DBA (set as the default attack) and ECBA. In DBA, the attacker decomposes a global 5×5 pixel white square trigger into four partial triggers (2×2, 2×3, 3×2, and 3×3), embedding them separately into the local data of different malicious clients. For ECBA, the attacker utilizes a fixed pool of 784 Southwest Airlines images [15] as the out-of-distribution (OOD) poison dataset. These images are globally shuffled and partitioned into mutually exclusive subsets across the malicious clients to simulate rare tail-data distribution. The class-based adaptive threshold is configured with a penalty coefficient of $\alpha = 16$ and a base threshold multiplier of $\beta = 8$. Note that all the above parameters serve as the default settings of our evaluation framework and are strictly applied unless explicitly stated otherwise.

B. Experiment 1: Analysis of the Transferability of Attacker Surrogate Models

Our framework uses ZKP commitments to prevent direct adaptive attacks against the actual reference model. Nevertheless, before submitting their ZKP commitments, attackers may still poison their local samples and generate evasive perturbations. Specifically, an attacker first trains a local surrogate model and exploits cross-model transferability to manipulate the cross-entropy loss distributions of poisoned and clean samples, with the aim of evading server-side auditing. To evaluate the framework against such adaptive attacks, we examine how the loss distributions of clean and poisoned samples change on both the surrogate models and the actual reference model.

The experiment is conducted on CIFAR-10 using ResNet-18. The local dataset submitted by the attacker to the pre-trained loss verification mechanism contains 10,000 samples, with poisoning implemented using a DBA trigger. The defender holds a clean reference dataset containing 500 samples. The attacker employs an adaptive evasion attack combining input diversity (DIM) [45] and momentum-based iterative optimization (MI-FGSM) [46]. Under a bounded perturbation budget, the attacker performs iterative gradient-based optimization on its local surrogate model to compress and reverse the loss separation between poisoned and clean samples.

We evaluate the attack under three threat scenarios based on the attacker's knowledge of the defender's reference dataset:

1) **Data-Unknown (S1):** The attacker uses 500 independent samples that do not overlap with the defender's reference dataset.
2) **Large-Scale (S2):** The attacker uses 5,000 independent samples that do not overlap with the defender's reference dataset.
3) **Exact-Known (S3):** The attacker uses exactly the same 500 reference samples as the defender but trains its model independently with different training randomness.

In practice, the attacker cannot access the defender's reference data. We include Scenario S3 solely to evaluate the theoretical upper bound of the attack.

During adaptive evasion, the attacker constructs a local surrogate model using the corresponding dataset to approximate the server's actual reference model. We use kernel density estimation (KDE) plots in Fig. 1 to illustrate the evolution of sample loss distributions on the local surrogate models and the actual reference model.

The right panel of Fig. 1 shows the loss density distributions evaluated on the attacker's local surrogate models. Before

adaptive evasion, the baseline poisoned samples, represented by the translucent red shaded region, exhibit higher losses than the baseline clean samples, represented by the translucent blue shaded region. Following local adaptive optimization, the relative positions of the distribution peaks are reversed: the red curves representing optimized poisoned samples shift substantially to the left, with their losses approaching zero, whereas the blue curves representing manipulated clean samples shift substantially to the right, indicating greatly increased losses. From the white-box perspective of the local surrogate models, the relative loss ordering of the two sample groups is reversed, allowing poisoned samples to exhibit lower losses than clean samples.

The left panel of Fig. 1 evaluates these adaptively perturbed samples on the actual reference model. Compared with the pronounced reversal observed on the surrogate models, the actual reference model exhibits substantially weaker changes. The attack induces a slight distribution shift, with the poisoned-sample loss distribution moving slightly to the left and reducing the separation between the two groups. However, the relative ordering of the clean and poisoned distributions is preserved. These perturbations are optimized for the local surrogate models, and their effects are substantially attenuated when transferred to the independently trained reference model. The remaining separation indicates that the local evasion effect does not transfer to the same extent to the defender's reference model.

Comparing the three threat scenarios reveals progressively stronger transfer effects, following the ordering $S1 < S2 < S3$. Transferability is weakest in S1, where the attacker uses a separate dataset, and increases when the attacker obtains more samples in S2 or the exact reference samples in S3. Nevertheless, even in S3, the perturbed distributions deviate only slightly from the baseline, and the relative loss ordering of clean and poisoned samples remains unchanged. These results indicate that, under the evaluated settings, surrogate-based adaptive evasion has a limited effect on the loss separation observed by the defender's reference model.

C. Experiment 2: Analysis of Different Combinations of Attack Parameters

After demonstrating robustness against black-box adaptive attacks, we evaluate its stability in gray-box settings. Since our reference model's dataset is relatively small, the reference model inevitably struggles to classify certain classes. The core question is whether an attacker can deliberately select these classes as poisoning targets, thereby bypassing loss-based detection.

To rigorously test the system's robustness against different attack combinations, we separately perform attacks on all 10 target classes in the CIFAR-10 dataset to eliminate any selection bias regarding class difficulty. At the same time, we increase the poisoning data rate from 0.1 to 0.4 to impose varying degrees of attack strength. The sample labeling results are shown in Fig. 2. (We omit the non-IID degree parameter in these visualizations because the spatial distribution of data across clients does not affect the absolute number of caught and missed samples captured in the global evaluation.)

Observing the stacked data distribution in the figure reveals a critical conclusion. Regardless of which difficult class the attacker chooses as camouflage, and regardless of how the poisoning rate fluctuates, the number of malicious samples successfully caught and forcibly relabeled by the system consistently and significantly exceeds the number of missed malicious samples. This numerical suppression forms the foundation of the betrayal mechanism. During the subsequent local training phase, a massive number of caught malicious samples are assigned the reference model's predicted labels and generate reverse gradients under the constraints of the ZKP protocol. Because these samples vastly outnumber the missed ones, their generated reverse gradients neutralize the backdoor updates carried by the few latent samples during model aggregation.

Furthermore, the total number of samples forcibly relabeled by the system consistently remains stable below 12,000. Compared with an ideal client-level detection mechanism with zero false positives, quarantining 30% of malicious nodes results in more samples being discarded. It should be pointed out that this only shows that our method has an advantage in quantity. It must be acknowledged that the impact of the relabeling method on the accuracy of the model's main task will ultimately be greater than the impact of directly discarding samples (we will discuss this impact in detail in Experiment 3).

D. Experiment 3: Analysis of Disposal Strategies for Flagged Samples

Experiment 2 has established that within the targeted classes, the sample-level auditing mechanism captures significantly more poisoned samples than it misses. Building upon this, the current experiment seeks to address two further questions: first, how should these captured samples be handled? Second, to what extent can the numerical advantage of captured samples over undetected ones be reduced while still maintaining the efficacy of the defense mechanism? This experimental section corresponds to the empirical analysis of Combinations 5 and 8 in Section IV-F.

We employ the default configuration of our evaluation framework, adjusting only two variables: the audit detection rate p (as defined in our game-theoretic model) and the disposal strategy for flagged samples. Under the default decision threshold, the detection rate is exceptionally high, resulting in almost zero undetected poisoned samples. Consequently, the residual backdoor payload faced by both disposal strategies approaches zero, making it difficult to observe any performance differences between them. To establish comparable experimental conditions, we deliberately increased the decision threshold to let some low-suspicion samples pass. This artificially reduced the detection rates to 90% and 80%, which corresponds to leaving approximately 10% and 20% of the poisoned samples undetected, respectively.

The two strategies differ only in how they handle flagged samples: the flipping strategy modifies their labels and keeps them in the training set, whereas the discarding strategy removes them from the dataset entirely. Importantly, under

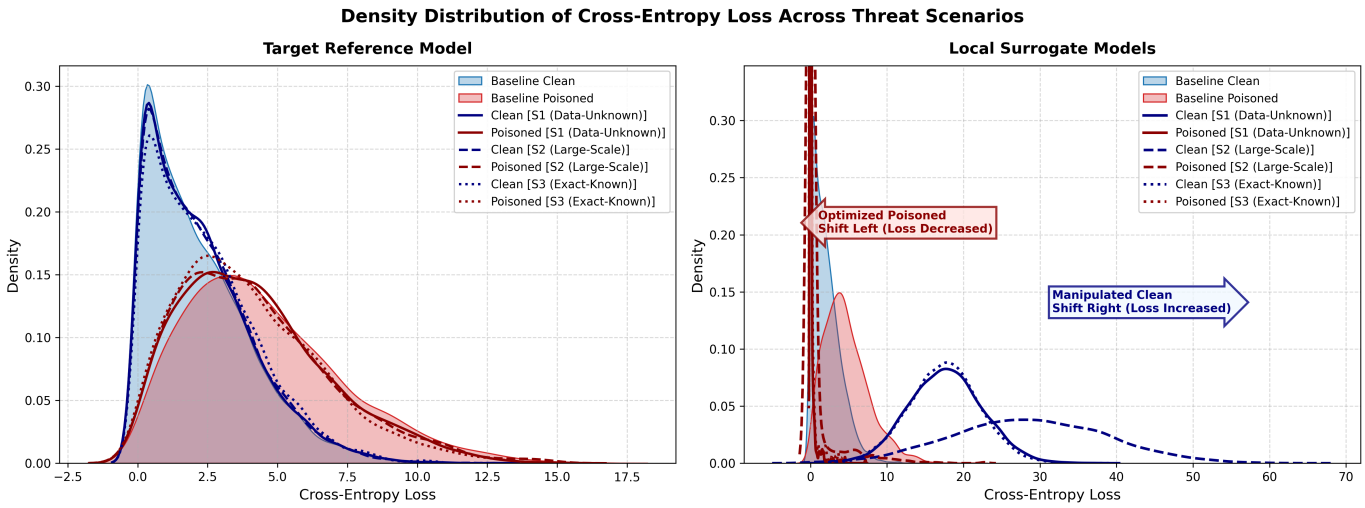


Fig. 1: Kernel density estimates of cross-entropy losses under different threat scenarios. **Right:** On the local surrogate models, adaptive optimization shifts the poisoned-sample distributions (red) to the left and the clean-sample distributions (blue) to the right, reversing their relative loss ordering. **Left:** On the actual reference model, the transfer effect increases in the order $S1 < S2 < S3$. Even in S3, the perturbed distributions deviate only slightly from the baseline, while the relative loss ordering of clean and poisoned samples is preserved.

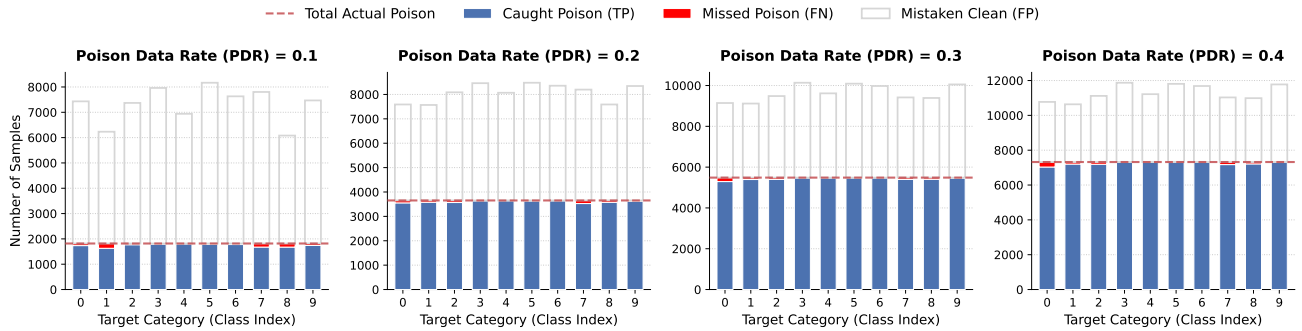


Fig. 2: Distribution of caught poisoned samples, missed poisoned samples, and mistakenly labeled clean samples across all target classes under varying poisoning data rates.

both strategies, the undetected poisoned samples remain in the training set with their triggers and target labels intact. Therefore, both strategies face the exact same backdoor threat; the only control variable is the disposal method applied to the flagged samples. Table II presents the detailed results of the auditing phase.

TABLE II: Audit output on the attacked class, together with the clean samples among the flagged ones.

Detection rate p	Caught : Missed	Flagged	Flagged (Clean)
90%	3286 : 367 ($\approx$ 9:1)	6333	3047
80%	2922 : 731 ($\approx$ 4:1)	5129	2207

Figure 3 reports the main task accuracy and attack success rate under four configurations. In terms of main task accuracy, the four curves overlap heavily, all rising rapidly in the early stage of training and then gradually converging. Nevertheless, there is still a slight difference between the two defense strategies: the flipping strategy slightly leads the discarding strategy in early training, the two cross in the middle, and ultimately the discarding strategy has a slightly higher accuracy. Regarding this anomalous phenomenon, we speculate that under the flipping strategy, some flagged poisoned samples happen to be restored to their true labels, thereby providing the model with additional benign data and giving it a slight early advantage. On the other hand, flagged clean samples are usually high-loss, hard-to-learn samples; because the model tends to prioritize fitting easy-to-learn samples in early training, flipping has a relatively low impact on accuracy. Overall, whether in early or late training, the difference between the two in main task accuracy is negligible and insufficient to serve as a basis for distinguishing these two strategies.

However, on the core metric of attack success rate, the two exhibit a significant difference. As shown in the figure, the red curves (discarding strategy) have an attack success rate far higher than the green curves (flipping strategy), indicating that the label flipping operation greatly suppresses the gradient updates generated by malicious samples. Moreover, even when the detection rate is reduced from 90% to 80%, this experimental result still holds. This finding strongly supports our

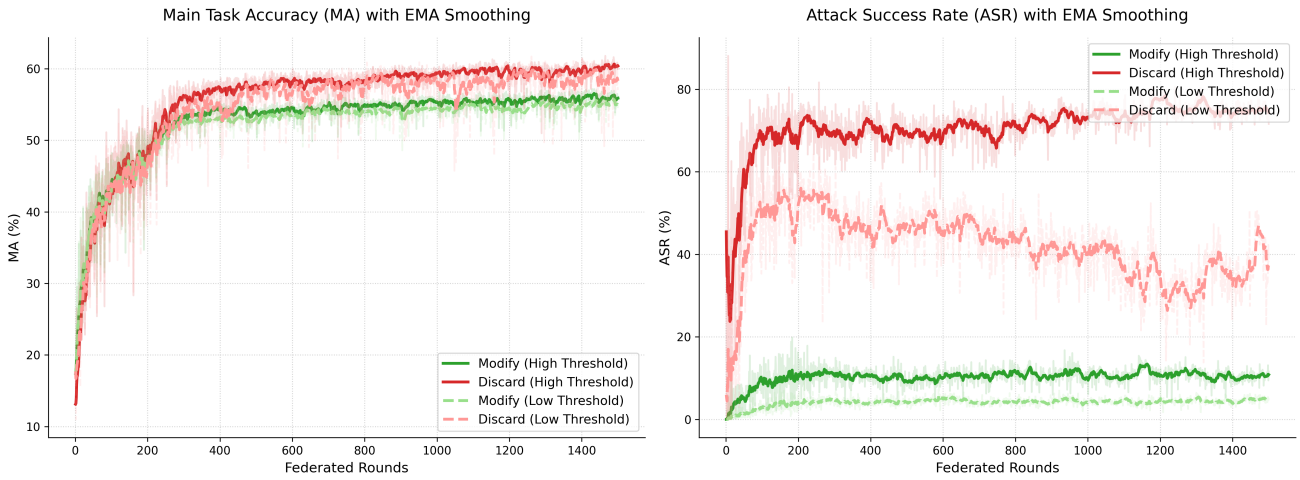


Fig. 3: Main task accuracy and attack success rate over federated rounds for the four configurations: flipping and discarding at detection rates of 90% and 80%. Thin lines are raw records and thick lines are exponentially smoothed.

conclusion: our defense framework does not need to perfectly identify all attack samples; by simply forcing the labels of suspicious samples to flip, it can cause these exposed attack samples to "betray" and offset the malicious influence of other potential undetected attack samples.

E. Experiment 4: Comparison with Existing Defenses under Extreme non-IID Settings

To evaluate the proposed method under extreme data heterogeneity, we compare it with four existing backdoor defenses within the same federated training framework against both DBA and ECBA. These defenses operate on distinct principles:

- **FoolsGold** [19]: Mitigates Sybil-based poisoning by evaluating the cosine similarity among client updates. It penalizes clients that submit highly similar gradients, operating on the assumption that malicious clients share a common attack objective and exhibit higher similarity than benign clients under non-IID settings.
- **FLTrust** [21]: Relies on a small, clean root dataset held by the server to compute a trusted reference update. We set the size of this dataset to be the same as that used by our method. FLTrust assigns aggregation weights based on the cosine similarity between client updates and the server's reference, while normalizing the magnitude of client updates to bound malicious impacts.
- **FLAME** [24]: Employs HDBSCAN clustering to identify and filter out anomalous client updates, followed by dynamic clipping and differential privacy noise injection to neutralize residual backdoor features.
- **FreqFed** [47]: Transforms model weights into the frequency domain and aggregates them based on frequency characteristics. It uses differences in frequency-domain representations to distinguish benign and malicious client models during aggregation.

Table III and Table IV report the MTA and the attack success rate (ASR) for DBA and ECBA, respectively, under Dirichlet partitions of $\mathrm{Dir}(0.1)$ and $\mathrm{Dir}(0.5)$.

TABLE III: MTA (%) and ASR (%) of the proposed method and four existing defenses against **DBA** under Dirichlet partitions of $\mathrm{Dir}(0.1)$ and $\mathrm{Dir}(0.5)$.

	$\mathrm{Dir}(0.1)$		$\mathrm{Dir}(0.5)$	
Method	MTA	ASR	MTA	ASR
FoolsGold	65.05	96.80	68.49	96.20
FLTrust	63.21	96.32	68.60	95.79
FLAME	48.67	78.63	65.51	81.86
FreqFed	51.64	93.62	61.74	95.33
Ours	55.65	5.96	61.20	1.73

TABLE IV: MTA (%) and ASR (%) of the proposed method and four existing defenses against **ECBA** under Dirichlet partitions of $\mathrm{Dir}(0.1)$ and $\mathrm{Dir}(0.5)$.

	$\mathrm{Dir}(0.1)$		$\mathrm{Dir}(0.5)$	
Method	MTA	ASR	MTA	ASR
FoolsGold	62.67	96.43	69.90	84.69
FLTrust	61.12	98.47	71.38	82.65
FLAME	53.53	97.96	64.56	77.55
FreqFed	52.16	89.80	61.49	70.41
Ours	55.49	5.61	60.16	4.08

As shown in Table III, all existing defenses almost completely fail against DBA, with ASRs remaining above 80% (and exceeding 93% for FoolsGold, FLTrust, and FreqFed). Notably, FLAME achieves a relatively lower ASR under $\mathrm{Dir}(0.1)$ (78.63%) than under $\mathrm{Dir}(0.5)$ (81.86%), yet this comes at the expense of a drop in MTA (plummeting to 48.67%). This indicates that FLAME's slight ASR reduction under extreme heterogeneity is not due to precise malicious update filtering, but rather because severe non-IID data triggers excessive dynamic clipping and differential privacy noise injection, which indiscriminately disrupts both backdoor features and main task convergence. These results demonstrate that under strong non-IID settings, gradient- and model-level defenses face a fundamental dilemma: the severe natural divergence

among benign client gradients completely masks the subtle gradient shifts induced by distributed micro-triggers.

In contrast, our method reduces the ASR to 5.96% (Dir(0.1)) and 1.73% (Dir(0.5)). Because our approach bypasses gradient inspection entirely and operates via sample-level analysis, the non-IID distribution across clients has virtually no impact on our sample flagging capability. The slight decrease in ASR from Dir(0.1) to Dir(0.5) should be interpreted as a result of more balanced benign local training under Dir(0.5), which yields a more robust global decision boundary that naturally dilutes residual unflagged backdoor features. Meanwhile, the data shows a moderate MTA reduction for our method compared to FoolsGold and FLTrust (whose higher MTAs are largely attributable to their defenses failing to effectively suppress malicious updates, leaving model training largely unaffected by defense intervention). Our MTA drop primarily stems from applying aggressive label flipping, rather than simple discarding, to the detected suspicious samples, which introduces minor label noise when benign samples are occasionally misflagged. We have thoroughly analyzed this trade-off in Experiment 3; notably, as demonstrated in Experiment 3, even when we intentionally adjust the threshold to admit more malicious samples and adopt a discarding strategy to preserve higher MTA, our ASR still remains drastically lower than all baseline defenses.

For ECBA (Table IV), existing defenses exhibit a moderate ASR reduction under Dir(0.5) compared to their complete failure against DBA. Specifically, when moving from Dir(0.1) to Dir(0.5), the ASRs drop from 96.43% to 84.69% for FoolsGold, 98.47% to 82.65% for FLTrust, 97.96% to 77.55% for FLAME, and 89.80% to 70.41% for FreqFed. This is because ECBA relies on OOD edge-case samples; unlike the imperceptible local triggers in DBA, mislabeling these unseen tail samples induces more distinctive gradient deviations and frequency-domain anomalies, allowing existing defenses to partially identify and suppress them when benign distributions are relatively consistent (Dir(0.5)). However, with ASRs still exceeding 70%, these defenses remain far from effective, and this distinctiveness is overwhelmed under Dir(0.1).

Conversely, our method consistently suppresses the ASR to 5.61% (Dir(0.1)) and 4.08% (Dir(0.5)), even though the reference model is trained exclusively on standard clean data without any exposure to ECBA samples. This effectiveness stems from two mechanisms. First, rather than searching for specific trigger patterns, the reference model simply verifies whether an image possesses the fundamental semantic features of its assigned label. Because an aircraft image, regardless of being OOD, completely lacks the core features of the target class, mislabeling it inevitably produces an anomalously high loss. Second, because this loss inspection is performed on individual samples rather than aggregated client gradients, the detection accuracy is entirely unaffected by the degree of client-level data heterogeneity, enabling subsequent label flipping to effectively neutralize the backdoor.

In summary, these comparisons confirm that existing defenses relying on cross-client gradient or frequency statistics struggle to survive extreme non-IID environments. By grounding detection in sample-level fundamental class features, our approach bypasses cross-client distribution skewness entirely and delivers robust protection against both DBA and ECBA under severe data heterogeneity.

VI. Discussion

A. Discussion on Clean-Label Attacks

Traditional federated learning defenses rarely evaluate clean-label poisoning specifically, as adversaries typically favor more destructive dirty-label poisoning or direct gradient manipulation. Under our framework, however, ZKP eliminates gradient manipulation and malicious scaling, while reference-model-based loss detection directly intercepts high-loss dirty-label samples caused by feature-label mismatches. Consequently, adaptive adversaries are forced to fall back on low-loss clean-label poisoning to evade screening.

Nevertheless, even if naive clean-label poisoning bypasses detection, it poses no substantive threat. Because these samples retain benign semantic features alongside correct labels, the model naturally relies on native features for classification during training (i.e., "feature competition"). Consequently, the gradients generated by poison triggers remain extremely weak; under strict constraints prohibiting gradient scaling, such faint updates are easily overwhelmed by benign updates during federated aggregation. Furthermore, if adversaries attempt to forcibly amplify poisoning potency via three mainstream enhancement strategies, each is precisely countered by our framework:

- **Adversarial Perturbations:** This approach injects adversarial perturbations prior to embedding the trigger, erasing the benign semantic features of the original image to eliminate feature competition [48]. However, this operation directly triggers our loss detection: trained exclusively on clean data, the server's reference model does not recognize the trigger. Once the native features are corrupted, the reference model fails to classify the sample into its ground-truth category, yielding an abnormally high loss that leads to immediate interception.
- **Feature Collision and Gradient Alignment:** Feature collision [49], [50] and gradient alignment [51], [52] seek to preserve visual appearance (maintaining low loss) while injecting carefully crafted subtle perturbations to align the features or gradients of poisoned samples with a malicious target. However, these methods rely heavily on prior knowledge of the target model parameters and require dynamically recomputing poisoned samples across communication rounds as the global model evolves. In our framework, clients must blindly commit to their local datasets via cryptographic commitments during initialization, prior to accessing either the reference model or the initial global model, thereby preventing adversaries from constructing or dynamically updating poisoned samples based on model parameters.
- **Hybrid Dirty-and-Clean Poisoning:** Adversaries may mix a small fraction of high-intensity "dirty-label" samples to force trigger learning, accompanied by a large volume of low-loss "clean-label" poisoned samples to evade detection and consolidate the backdoor. Under

conventional filtering mechanisms, even if the few dirty samples are discarded, the remaining clean-label poisoned samples can still sustain the attack. Conversely, rather than simply discarding detected dirty samples, our framework forcibly relabels them to the true categories predicted by the reference model. Consequently, the captured dirty samples retain their triggers while being bound to ground-truth labels, generating strong corrective gradients during local training that directly counteract and neutralize the consolidating effect of the unflagged clean-label poisoned samples.

B. Discussion on Scalar Loss Privacy

A natural concern is whether reporting low-dimensional scalar losses in Phase 2 compromises client privacy. Counter-intuitively, exposing these 1D scalars enables strictly stronger end-to-end privacy guarantees than conventional defenses.

This advantage stems from the fundamental difference in invertibility between a 1D scalar loss and a high-dimensional gradient vector. Evaluating a local sample on the reference model projects high-dimensional input features into a single scalar value. Because this many-to-one mapping is mathematically non-invertible, an adversary cannot reconstruct raw private features from isolated loss values. In contrast, high-dimensional gradients carry rich spatial and semantic information across millions of parameters. Gradient inversion attacks such as DLG [53] and Inverting Gradients [54] have proven that an adversary can reconstruct private training images with pixel-level fidelity directly from unmasked updates. Despite this vulnerability, mainstream defenses [19], [21], [24] require the server to inspect individual plaintext gradients in every training round to compute non-linear distance metrics or filter anomalies. This requirement creates an architectural conflict with Secure Aggregation [55], which cryptographically conceals individual updates to prevent reconstruction attacks but simultaneously blinds the server and paralyzes these gradient-based defenses.

Our architecture resolves this privacy-robustness dilemma at the design level by confining anomaly detection to a one-time inspection of low-dimensional scalar losses in Phase 2. Because backdoor mitigation is completed before formal training begins, the server requires zero visibility into individual local gradients and only needs to perform standard linear aggregation. Although we do not explicitly implement Secure Aggregation in this work, this decoupling makes our training natively compatible with commitment-based SecAgg frameworks such as RoFL [33], which utilize ZKPs to verify update constraints directly over homomorphically masked commitments without exposing individual gradients. By trading a one-shot, non-invertible scalar loss for architectural compatibility with encrypted gradient aggregation across all training rounds, our approach paves the way for achieving simultaneous backdoor robustness and strong gradient confidentiality in future deployments.

VII. Limitations

A. Limitations of the Reference Model

A core limitation of our screening mechanism is its reliance on a server-side reference dataset. While requiring only 500 clean images (merely 1% of client data), allowing the server to hold any task-specific raw data inherently relaxes the strict data-isolation premise of standard federated learning, potentially facing regulatory hurdles in privacy-critical domains. Moreover, this compact reference dataset limits the coverage of rare benign features, occasionally causing false-positive relabeling on atypical clean samples and slightly degrading main-task accuracy.

Future work can address these limitations practically. First, to eliminate server-side raw data, the reference model can be initialized using synthetic proxy data or trained collaboratively across audited consortium nodes via secure aggregation. Second, to prevent false-positive relabeling, the screening rule can be upgraded to require both a high loss on the original label and a sufficiently high confidence score on the new predicted class.

B. Limitations of zkML and System Deployment

Furthermore, the integration of zero-knowledge proofs and deposit slashing introduces practical system constraints. Generating and verifying proofs for high-dimensional gradients and large local datasets requires resources that typically exceed the capabilities of standard edge devices, while strict penalties against disconnections may inadvertently affect honest clients experiencing transient network failures. Consequently, our framework is immediately suited for cross-silo federated consortia with dedicated infrastructure and stable connectivity. To extend the system to resource-constrained cross-device environments, future work can incorporate reconnection grace periods alongside optimized zkML operators, hardware acceleration, and probabilistic verification.

VIII. Conclusion

In this paper, we presented a backdoor defense for federated learning that inspects individual data samples before training rather than filtering aggregated client gradients. By requiring clients to commit to their local datasets before the reference model is revealed and verifying both loss evaluations and training updates with zero-knowledge proofs, the system prevents attackers from tampering with gradients, swapping data between stages, or launching adaptive attacks against the reference model. Because sample-level loss on the reference model depends only on whether an image matches its assigned label, our framework reliably catches poisoned samples regardless of non-IID client distributions or split triggers. More importantly, instead of simply discarding flagged data, forcing clients to relabel suspicious samples under a deposit penalty turns captured poison samples into reverse gradients that actively cancel out unflagged ones during training, allowing the system to neutralize backdoors without requiring a perfect detection rate.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. Pmlr, 2017, pp. 1273–1282.

[2] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[3] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," *Google Research Blog*, vol. 3, 2017.

[4] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[5] Y. Wan, Y. Qu, W. Ni, Y. Xiang, L. Gao, and E. Hossain, "Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 1861–1897, 2024.

[6] P. Ye, Y. Li, K. He, Q. Li, T. Qin, X. Wang, K. Yang, C. Zhang, and J. Chen, "Breaking the illusion: A critical study of backdoor defense in federated learning with non-iid data," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 13 264–13 279, 2025.

[7] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[8] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[9] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[10] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International conference on machine learning*. PMLR, 2019, pp. 634–643.

[11] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Ndss*, 2021.

[12] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[13] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," in *Uncertainty in artificial intelligence*. PMLR, 2020, pp. 261–270.

[14] 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.

[15] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in neural information processing systems*, vol. 33, pp. 16 070–16 084, 2020.

[16] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[17] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning." *ICLR*, vol. 2020, pp. 1–19, 2020.

[18] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[19] ——, "The limitations of federated learning in sybil settings," in *23rd International symposium on research in attacks, intrusions and defenses (RAID 2020)*, 2020, pp. 301–316.

[20] 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.

[21] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020.

[22] 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, pp. 5650–5659.

[23] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd annual conference on computer security applications*, 2016, pp. 508–519.

[24] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, S. Zeitouni *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX security symposium (USENIX Security 22)*, 2022, pp. 1415–1432.

[25] P. Rieger, T. D. Nguyen, M. Miettinen, and A.-R. Sadeghi, "Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection," *arXiv preprint arXiv:2201.00763*, 2022.

[26] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," *arXiv preprint arXiv:1909.05125*, 2019.

[27] T. Minka, "Estimating a dirichlet distribution," 2000.

[28] 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.

[29] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 2545–2555.

[30] P. Rieger, T. Krauß, M. Miettinen, A. Dmitrienko, and A.-R. Sadeghi, "Crowdguard: Federated backdoor detection in federated learning," *arXiv preprint arXiv:2210.07714*, 2022.

[31] S. Han, W. Wu, B. Buyukates, W. Jin, Q. Zhang, Y. Yao, and S. Avestimehr, "Kick bad guys out! conditionally activated anomaly detection in federated learning with zero-knowledge proof verification," *arXiv preprint arXiv:2310.04055*, 2023.

[32] Y. Chang, Q. Chen, T. H. Luan, and Z. Su, "Rankfl: Robustness and privacy-preserving federated learning scheme against poisoning attacks," *IEEE Transactions on Mobile Computing*, 2026.

[33] H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, and A. Hithnawi, "Rofl: Robustness of secure federated learning," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 453–476.

[34] D. Gupta, "Privacy-preserving federated learning: Integrating zero-knowledge proofs in scalable distributed architectures," *arXiv preprint arXiv:2605.08152*, 2026.

[35] T. Nguyen and M. T. Thai, "Preserving privacy and security in federated learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 833–843, 2023.

[36] J. Zhang, X. Bai, and Q. Zhang, "Efficient and privacy-preserving federated learning with gradient control against data poisoning," *IEEE Transactions on Consumer Electronics*, 2026.

[37] C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang, "Mystique: Efficient conversions for {Zero-Knowledge} proofs with applications to machine learning," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 501–518.

[38] M. Hao, H. Chen, H. Li, C. Weng, Y. Zhang, H. Yang, and T. Zhang, "Scalable zero-knowledge proofs for non-linear functions in machine learning," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 3819–3836.

[39] C. Weng, K. Yang, J. Katz, and X. Wang, "Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1074–1091.

[40] K. Yang, P. Sarkar, C. Weng, and X. Wang, "Quicksilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field," in *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, 2021, pp. 2986–3001.

[41] T. Liu, X. Xie, and Y. Zhang, "Zkcnn: Zero knowledge proofs for convolutional neural network predictions and accuracy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2968–2985.

[42] H. Duan, Z. Peng, L. Xiang, Y. Hu, and B. Li, "A verifiable and privacy-preserving federated learning training framework," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 5, pp. 5046–5058, 2024.

[43] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[45] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. IEEE, 2019, pp. 2725–2734.

[46] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 9185–9193.

[47] H. Fereidooni, A. Pegoraro, P. Rieger, A. Dmitrienko, and A.-R. Sadeghi, "Freqfed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning," *arXiv preprint arXiv:2312.04432*, 2023.
[48] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.
[49] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Advances in neural information processing systems*, vol. 31, 2018.
[50] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 957–11 965.
[51] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein, "Witches' brew: Industrial scale data poisoning via gradient matching," *arXiv preprint arXiv:2009.02276*, 2020.
[52] H. Souri, L. Fowl, R. Chellappa, M. Goldblum, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 165–19 178, 2022.
[53] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.
[54] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.
[55] 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.