CloakFed: Exploring Stealthy and Adaptive Backdoor Attacks in Federated Learning

Kaibo Wang, Anqi Zhang, Tangyou Liu, Wenqian Zhang*, and Guanglin Zhang*
College of Information Science and Technology, Donghua University, Shanghai, China
{2222080, aqzhang}@mail.dhu.edu.cn, {liuty, wqzhang, glzhang}@dhu.edu.cn

***Abstract*—Federated Learning (FL) is a privacy-preserving distributed machine learning paradigm that enables participants to collaboratively train a global model without sharing local private data. However, the distributed nature of FL introduces the potential risk of backdoor attacks, where malicious participants manipulate the model by injecting trigger patterns during local training, causing the global model to make incorrect predictions for specific inputs. Traditional backdoor attacks usually involve fixed-pattern triggers, which are vulnerable to detection or filtering in subsequent validation stages. Moreover, these attacks often fail to bypass the complex defense mechanisms employed during the model training phase on the server. To address these issues, we propose a stealthy and adaptive FL backdoor attack framework, CloakFed, which aims to hide all traces of the backdoor attack. In the data processing phase, CloakFed uses a stealthy generator to produce trigger patterns with hidden features similar to target labels, making the poisoned samples more similar to normal data in distribution, thus enhancing stealth and reducing the risk of manual detection. In the backdoor training phase, CloakFed inspects hook return values to infer potential defense strategies employed by the server and adaptively selects appropriate attack methods, such as model dimension attacks or model space attacks. This flexible strategy enables CloakFed to effectively execute backdoor attacks even in the presence of diverse defense mechanisms. To validate CloakFed's effectiveness, we conducted extensive experiments on three different datasets. Experimental results demonstrate that CloakFed achieves a significantly higher attack success rate under various defenses compared to three existing classic FL backdoor attacks. Specifically, in fixed-frequency and few-shot attack scenarios, CloakFed exhibits superior stability and persistence.**

***Index Terms*—Federated learning, backdoor attack, trigger patterns, model dimension attack, model space attack**

I. Introduction

Federated Learning (FL) is a distributed machine learning paradigm that enables multiple clients to collaboratively train a global model without sharing their private data. FL has been widely adopted in various applications, such as healthcare [1], Internet of Things [2], and edge computing [3]. However, FL is susceptible to backdoor attacks, where malicious clients insert hidden backdoors into the local model during training. These backdoors are triggered by specific patterns, causing the global model to make incorrect predictions when exposed to the triggers.

This work was supported in part by the National Natural Science Foundation of China under Grant 62301307, in part sponsored by Shanghai Pujiang Programme, in part by the Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission.

Existing FL backdoor attacks can be categorized into two types: data poisoning attacks and model poisoning attacks. Data poisoning attacks typically rely on fixed-pattern triggers or optimize trigger patterns using models trained on client-local data [4]. However, these methods have two major drawbacks: fixed-pattern triggers increase the risk of detection due to abnormal model modifications [5, 6], while attacks effective on local models may not transfer well to the global model due to heterogeneity [7]. In model poisoning attacks, attackers often need sufficient system prior information [8], such as the structure of the global model, learning rate, and clipping norms [9], in order to evade detection when the server equipped with defense mechanisms. However, this is impractical in real-world scenarios.

We propose CloakFed, a novel FL backdoor attack framework that enhances attack stealth and effectiveness by simultaneously optimizing data poisoning and model poisoning attacks. CloakFed consists of two key components: a stealthy trigger generator and adaptive attack strategies. The stealthy trigger generator utilizes concepts from image steganography and deep neural networks to create sample-specific invisible additive noise as backdoor triggers. This approach significantly improves the stealthiness of the triggers, making them difficult to detect. The generator is trained using a combination of backdoor loss and visual loss, striking a balance between attack effectiveness and visual imperceptibility. The adaptive attack strategies are implemented through a hook module insertion mechanism and model training strategy modifications. The hook module tracks the effectiveness of backdoor attacks by monitoring critical computation nodes in the global model, providing feedback for adjusting attack strategies. Based on this feedback, CloakFed can dynamically adapt its attack approach, including model-dimension and model-space attacks, to evade various defense mechanisms employed by the server. By integrating these components, CloakFed offers a comprehensive framework that can maintain attack effectiveness while adapting to different FL defense strategies, thus presenting a more resilient and stealthy backdoor attack method in federated learning environments.

The main contributions of this paper are as follows:

- CloakFed is the first backdoor attack framework that combines a stealthy trigger generator with adaptive attack strategies to launch stealthy and resilient FL backdoor attacks in various defense scenarios.
- We incorporate a novel hook module insertion mechanism that enables the attacker to obtain critical feedback on the effectiveness of their attacks and the potential defense

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

Algorithm 1: CloakFED

Input: Local dataset D_i, Global model g^{t-1}, Target label y_{target}, Epoch of backdoor attack $e_{backdoor}$, Trigger generator ξ^{t-1}, Critical computation nodes Index $\hat{M}^{t-1}$

Output: Backdoor model w_i^t, critical computation nodes Index $\hat{M}^t$ and backdoor dataset $\hat{D}_i$

1: **for** epoch $e \in \{1 \dots e_{backdoor}\}$ **do**
2: ▷ Data processing phase
3: $\xi^t \leftarrow$ Stealthy Trigger Generation(D_i, g^{t-1}, ξ^{t-1}, y_{target})
4: ▷ Backdoor model training phase
5: $w_i^t, \hat{D}_i \leftarrow$ Backdoor Model Training(D_i, g^{t-1}, ξ^t, $\hat{M}^{t-1}$)
6: $\hat{M}^t \leftarrow$ Hook Module Insertion(w_i^t, g^{t-1})
7: **end for**
8: **return** $w_i^t, \hat{D}_i, \hat{M}^t$

strategies employed by the server.

- Extensive experiments across three benchmark datasets demonstrate CloakFed's superior attack success rates and stealthiness against six federated learning defense mechanisms.

II. Preliminary

Federated Learning. Consider a FL system with N clients in total, n^t clients are selected for the current training round. The central server distributes the global model g^{t-1}, to the selected clients. Each client then optimizes the model using its local data. For example, using the Stochastic Gradient Descent (SGD) method, the local update of client i in round t can be expressed as:

$$w_i^t \leftarrow g^{t-1} - \gamma \nabla \mathcal{L}_{task}, \tag{1}$$

where γ is the local learning rate, and $\nabla \mathcal{L}_{task}$ denotes the gradient of the task-specific loss function. Upon completing local training, clients send the updated model parameters back to the central server. The server then aggregates the updates from all clients and generates a new global model:

$$g^t = g^{t-1} + \frac{\eta}{n^t} \sum_{i=1}^{n^t} (w_i^t - g^{t-1}), \tag{2}$$

where η is the global learning rate. This process improves the global model while preserving the data privacy of each client.

III. Method

A. Stealthy Trigger Generation

Inspired by the process of training a Generative Adversarial Networks (GANs), we introduce an innovative method in the data processing stage to generate a more stealthy trigger. As shown in Fig. 1, we employ the global model provided by the server as the discriminator and train the trigger generator $\mathcal{T}_{\xi^t}$ by combining attacker-specified string encoding with benign

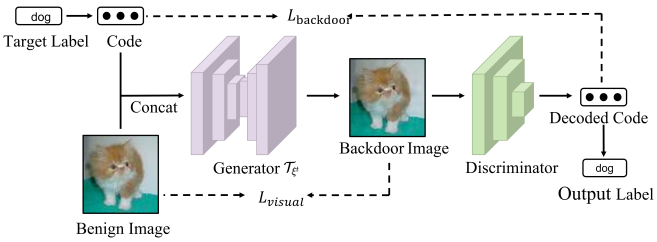

Fig. 1: The Stealthy Trigger Generator Training Process.

images. The objective of the generator is to create sample-specific invisible additive noise as the backdoor trigger δ. This design markedly improves the stealthiness of the trigger and makes it difficult to be detected. Under this framework, the backdoor sample $B_{\xi^t}(x)$ can be formally modeled as:

$$B_{\xi^t}(x) = x + \delta, \tag{3}$$

$$\delta = \mathcal{T}_{\xi^t}(x, y_{target}), \forall x \in \mathcal{D}_i, \tag{4}$$

where x denotes any benign sample belonging to the local dataset D_i, y_{target} represents the target label, and ξ^t denotes the trigger generation parameter.

To ensure that backdoor samples remain visually indistinguishable from benign samples while simultaneously misleading the discriminator, we must further optimize the training of the generator. Previous studies [10, 11] have constrained the l norm of the trigger δ within a specified range ϵ, i.e., $||\delta||_2 \leq \epsilon$ or $||\delta||_\infty \leq \epsilon$. By adjusting the ϵ value, it is possible to regulate both the size of the trigger and the magnitude of pixel alterations, thereby achieving the desired balance between visual stealth and attack effectiveness. However, our research indicates that solely relying on a fixed ϵ value to limit the l norm of the trigger introduces inherent limitations.

As illustrated in the Fig. 2, although a larger ϵ can significantly increase the effectiveness of the backdoor attack, it compromises visual stealth, making the trigger more easily detectable by the human eye and thus reducing the attack's overall stealthiness. Conversely, a smaller ϵ can preserve the visual stealth of the trigger, but it often leads to a marked decrease in attack effectiveness, particularly in federated learning environments, where the global model is less likely to react to such subtle changes.

To address the limitations of the fixed ϵ mentioned above, we introduce an image visual loss $\mathcal{L}_{\text{visual}}$ to enhance the steganographic performance of the generative model. Traditional l-norms assess pixel-level differences, but these often do not align perfectly with human visual perception. In contrast, the feature layers of deep neural networks can better capture high-level semantic information in images, making them more aligned with human perception when evaluating image similarity. Specifically, we use the VGG network as an evaluation model and incorporate the Learned Perceptual Image Patch Similarity (LPIPS) as the loss function to construct $\mathcal{L}_{\text{visual}}$. The LPIPS loss measures the similarity between the original image x and the backdoor image $B_{\xi^t}(x)$ by calculating feature differences

across multiple convolutional layers as

$$\mathcal{L}_{\text{visual}}(x, B_{\xi^t}(x)) = \sum_l w_l \cdot |\phi_l(x) - \phi_l(B_{\xi^t}(x))|, \quad (5)$$

where ϕ_l denotes the feature map of the l-th layer of the VGG network, and w_l represents the weight of the l-th layer.

To evaluate the practical impact of the generated backdoor samples on the global model, we employ the cross-entropy loss function $\mathcal{L}_{\text{backdoor}}$ to measure the effectiveness of the backdoor trigger on the attack target. By considering both visual stealth and attack effectiveness, we construct the trigger generator's overall loss function as follows:

$$\mathcal{L}_{\text{trigger}} = \mathcal{L}_{\text{backdoor}} + \mathcal{L}_{\text{visual}}. \quad (6)$$

By optimizing the trigger generator based on the combined loss function $\mathcal{L}_{\text{trigger}}$, we can effectively enhance the stealthiness of the trigger while maintaining the attack's effectiveness.

B. Hook Module Insertion

Motivated by the work in [12], we introduce a mechanism called the hook module in CloakFed, designed to track the effectiveness of backdoor attacks and provide feedback for adaptive defense strategies. The core of this mechanism is implemented using dynamic instrumentation techniques. Specifically, the attacker identifies and searches for the bottom-$m\%$ of critical computation nodes within the global model, represented by an index map M, and executes hook operations at these nodes. At the onset of backdoor training, the hook module reads and returns relevant values, aiding in the assessment of whether the backdoor samples have been accepted by the global model. Based on this feedback, the backdoor training strategy can be dynamically adjusted.

The selection of critical computation nodes is based on two criteria:

C1: The selected computation nodes should experience significantly smaller gradient updates during training compared to other nodes.

C2: When the parameters of these computation nodes undergo slight changes, ensuring that the task's loss does not significantly increase.

Algorithm 2: Hook Module Insertion

Input: Global model g^{t-1}, Backdoor model w_i^t
Output: Critical computation nodes Index $\hat{M}^t$

1: $G^t \leftarrow \nabla\mathcal{L}_{task}(g^{t-1}, D_i)$ ▷ Compute Gradient
2: $M = \{M_1, M_2, ..., M_j\} \leftarrow \arg\min_{M_i} \theta_{G^t, M_i}$
3: $H^t \leftarrow \nabla^2\mathcal{L}_{task}(g^{t-1}, D_i)$ ▷ Compute Hessian
4: $\hat{M}^t = \{\hat{M}_1, \hat{M}_2, ..., \hat{M}_j\} \leftarrow arg\min_{\hat{M}_i \in M} \theta_{H^t, \hat{M}_i}$
5: **return** $\hat{M}^t$

As shown in Algorithm 2, we take the global model g^{t-1} and the local dataset D_i as inputs, and based on the loss function $\mathcal{L}_{task}$, we compute the approximate gradient of the global model parameters G^t. For each computation node, we sort the absolute values of the gradients and select the indices of the parameters with the smallest gradient magnitudes $\arg\min_{M_i} \theta_{G^t, M_i}$. These indices are then recorded in the map $\{M_1, M_2, ..., M_j\}$, ensuring that the condition of small gradient update magnitudes is satisfied.

The Hessian matrix of the loss function indicates the direction of gradient updates. In other words, if the value of the second-order derivative is sufficiently small, even large adjustments to that parameter will not significantly affect the loss function. Therefore, we further select the parameter index map H^t corresponding to the smallest curvature $\arg\min_{\hat{M}_i \in M} \theta_{H^t, \hat{M}_i}$, ensuring that changes at the computation nodes do not significantly affect the task's performance.

After completing local training, we multiply the parameter update at index $\hat{M}_i$ by an adaptive factor k ($k > 1$) to enhance the effect of the backdoor attack. Meanwhile, we record the adjusted parameter update $\theta_{w_i^t - g^{t-1}, \hat{M}_i}$, and compare it with the parameter update at the same index $\hat{M}_i$ in the next round of the global model, thereby inferring potential defense strategies that the server might adopt.

Inferring potential defense strategies depends on the discriminative metric I. The formula for calculating the discriminative metric is as follows:

$$I_i = \left| \frac{\theta_{g^t - g^{t-1}, \hat{M}_i}}{\theta_{w_i^t - g^{t-1}, \hat{M}_i}} \right|, \quad (7)$$

where $\theta_{g^t - g^{t-1}, \hat{M}_i}$ represents the parameter updates of the global model at the index map $\hat{M}$. The attacker then compares the discriminative metric I_i with a predefined threshold to adjust subsequent backdoor training strategies.

C. Model Training Strategies

As previously mentioned, we fixed the generator's parameters to train the local model. In this process, the local model needs to perform well in two aspects: on the one hand, it must correctly classify benign samples, which can be achieved by minimizing the loss function $\mathcal{L}_{clean}(w_i^t, x, y)$; on the other hand, it must misclassify samples with backdoor triggers into the attacker's predefined target label y_{target}, which can be ensured by minimizing $\mathcal{L}_{backdoor}(w_i^t, B_\xi(x), y_{target})$. Therefore, this task can be formulated using the following objective function $\mathcal{L}_{task}$:

$$\mathcal{L}_{task} = \alpha \cdot \mathcal{L}_{clean} + (1 - \alpha) \cdot \mathcal{L}_{backdoor}, \quad (8)$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the trade-off between correctly classifying benign samples and successfully executing the backdoor attack.

Model-dimension attack To enhance the stealth and persistence of the backdoor attack, we choose to target the critical computation nodes identified in the Section III-B. These nodes are rarely covered or corrected by the gradient updates from other benign clients, which not only extends the duration of the backdoor effect but also reduces the risk of being detected by defense mechanisms. In our experiments, we set $m = 5$. As training progresses and most clients converge to the global model, the bottom critical computation nodes identified in each round become increasingly stable.

Model-space attack Defense schemes based on noise perturbation often require gradient norm clipping. In FL backdoor attacks, attackers amplify their gradients to counteract the normal gradients of other benign clients. Research [13]

Algorithm 3: Backdoor Model Training

Input: Local dataset D_i, Global model g^{t-1}, Critical computation nodes Index $\hat{M}^{t-1}$, Trigger generator ξ^t
Output: Backdoor model w_i^t and backdoor dataset $\hat{D}_i$

Generate backdoor dataset $\hat{D}_i$ with fixed $\mathcal{T}_{\xi^t}$
for each parameter index $\hat{M}_i \in \hat{M}^{t-1}$ **do**
  Calculate the discriminative metric through Eq.(7)
  if $I_i > 1$ **then**
    Optimize w_i^t by using SGD though Eq.(9)
  else if $I_i \leq \frac{1}{k}$ **then**
    Optimize w_i^t by using SGD though Eq.(8)
  else
    Optimize w_i^t by using SGD with though Eq.(8)
    for each parameter index $\hat{M}_i \in \hat{M}^{t-1}$ **do**
      $\theta'_{w_i^t - g^t, \hat{M}_i} \leftarrow k \cdot \theta_{w_i^t - g^t, \hat{M}_i}$
    end for
  end if
end for
return $w_i^t, \hat{D}_i$

shows that when the server applies a dynamic clipping bound for norm clipping during each gradient update, the effectiveness of backdoor attacks that rely on gradient amplification can be significantly mitigated.

To resist norm clipping, we propose adding a model norm space constraint to the loss function:

$$\mathcal{L}_{task} = (1-\lambda) \cdot \mathcal{L}_{task} + \lambda \cdot ||w_i^t - g^{t-1}||_2, \qquad (9)$$

where λ is a hyperparameter that balances the trade-off between the task loss and the model norm space constraint.

IV. Experiments

A. Experimental Setup

Datasets and System setup. Our experiments utilized three real-world image classification datasets: FEMNIST [14], CIFAR-10 [15], and Tiny-ImageNet [16]. For the FEMNIST dataset, we conducted experiments using a simple 2-conv-2-fully-connected model with a learning rate of 0.1, while for CIFAR-10 and Tiny-ImageNet, we applied the ResNet-18 [17] architecture and set the learning rate to 0.01.

To simulate non-IID datasets, we applied the Dirichlet distribution to partition each client's dataset. For the MNIST dataset, we used $Dir_k(0.5)$, whereas for the CIFAR-10 and Tiny-ImageNet datasets, we used $Dir_k(0.1)$. By default, we set the number of clients 100. During each communication round, the server randomly selects 10 clients to participate in the aggregation process.

Attack settings. To evaluate the stability and robustness of CloakFED under different attack scenarios, we considered two typical attack methods: fixed-frequency attacks and few-shot attacks.

Baselines. To conduct a systematic comparative study, we compared CloakFED with three state-of-the-art attack methods: Iba [11], Neurotoxin [18], and Edge-case [9]. At the same time, we introduced six classical or advanced federated learning defense mechanisms for evaluation: Krum [19], Flame [20], Deepsight [13], Foolgold [21], SparseFed [22] and Robust Federated Aggregation (RFA) [23].

Evaluation metrics. Building on previous works [9, 24, 25, 18], we evaluate the effectiveness of the backdoor attack using metrics such as Visual Imperceptibility, Main Accuracy (MA), Attack Success Rate (ASR), and Lifespan.

B. Experimental Results

TABLE I: The Relationship Between Visual Imperceptibility and Attack Success Rate in Three Attack Schemes. In Iba, visual imperceptibility is altered by controlling the magnitude of the l_∞-norm.

Dataset	FEMNIST			CIFAR-10			Tiny-ImageNet		
	PSNR↑	SSIM↑	ASR(%)	PSNR↑	SSIM↑	ASR(%)	PSNR↑	SSIM↑	ASR(%)
Neurotoxin	26.43	0.948	95.83	25.91	0.926	82.47	24.54	0.899	88.16
Iba($\epsilon = 0.001$)	42.63	0.996	73.24	41.76	0.995	58.65	41.68	0.994	63.90
Iba($\epsilon = 0.05$)	37.56	0.978	90.65	36.56	0.972	74.56	37.39	0.964	83.73
Iba($\epsilon = 0.10$)	31.19	0.927	98.12	29.93	0.897	86.71	28.16	0.902	87.43
CloakFED	37.48	0.985	97.31	35.30	0.973	87.82	36.28	0.967	92.51

Visual Imperceptibility Comparison. The Fig. 2 compares different triggers on CIFAR-10 images. By comparing the differences between the first and second rows, it can be observed that CloakFED excels in maintaining high consistency with the visual characteristics of benign images, making it difficult for human observers to detect. By analyzing the residual images, it is found that CloakFED primarily embeds triggers by focusing on the structure of the object (such as outlines and edges) without altering the original texture of the image, thereby minimizing the impact on visual effects. When the upper bound is set to $\epsilon = 0.05$, Iba's visual imperceptibility is comparable to that of CloakFED, but its ASR significantly decreases. As ϵ is further reduced, the drop in ASR becomes more pronounced. Combined with the data in Table I, it can be seen that when $\epsilon = 0.10$, Iba's ASR approaches that of CloakFED, but this comes at the expense of its visual imperceptibility. In subsequent experiments we use $\epsilon = 0.05$.

Stability of CloakFED under Fixed-Frequency Mode. We plotted Fig. 3 and Fig. 4 on the CIFAR-10 dataset, with additional experimental results listed in Table II. From Fig. 3, it can be observed that when the server implements no defense mechanism, Neurotoxin, Iba, and CloakFED all achieve high ASR, while the MA on benign samples remains within 5%

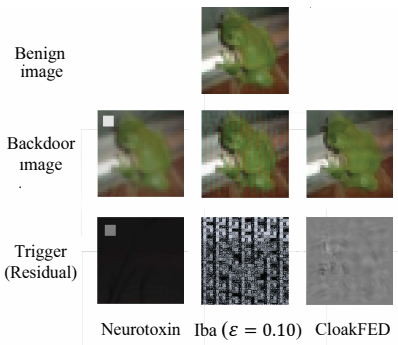

Fig. 2: Trigger Comparison.

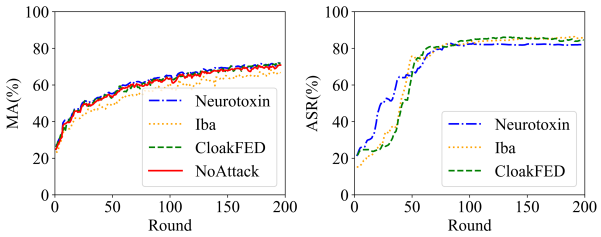

Fig. 3: MA/ASR of Iba, Neurotoxin, and CloakFED under FedAvg.

TABLE II: The ACC/ASR(%) Performance of Various Backdoor Attacks Against Different Defense Strategies.

Datasets	Attacks	FedAvg	Krum	Flame	Deepsight	Foolgold	SparseFed	RFA
FEMNIST	NoAttack	83.19/-	71.55/-	81.74/-	82.31/-	83.41/-	76.94/-	83.70/-
	Neurotoxin	79.23/95.83	68.53/94.17	80.27/57.27	79.37/81.42	83.48/88.03	76.18/24.51	80.78/27.75
	Iba	81.94/90.65	70.36/92.64	80.02/37.91	80.65/87.10	81.97/39.32	77.30/85.72	83.52/46.30
	CloakFED	81.52/96.31	71.45/98.37	82.39/91.68	81.54/94.23	83.69/94.61	76.26/87.19	83.41/93.18
CIFAR-10	NoAttack	70.63/-	57.27/-	70.11/-	68.23/-	70.40/-	62.99/-	70.43/-
	Neurotoxin	64.26/82.47	53.11/81.63	68.29/13.79	69.24/61.48	69.26/79.46	60.77/9.84	72.18/9.74
	Iba	70.69/84.56	59.45/82.55	70.56/22.35	65.29/75.92	69.76/10.39	61.21/72.53	70.09/17.32
	CloakFED	69.41/87.82	60.76/86.91	69.48/81.23	67.49/81.54	70.33/82.07	61.52/76.42	71.47/82.95
Tiny-ImageNet	NoAttack	41.92/-	27.73/-	39.32/-	39.63/-	43.10/-	32.77/-	43.62/-
	Neurotoxin	40.33/87.16	18.26/87.28	37.29/6.27	38.41/73.44	41.64/79.38	31.08/14.62	41.58/8.39
	Iba	40.70/83.73	23.07/84.91	36.56/15.83	39.15/81.95	42.98/8.23	32.59/65.83	40.82/17.82
	CloakFED	41.31/92.51	26.19/91.35	39.05/82.30	39.72/87.28	43.72/84.11	32.11/79.76	42.83/89.02

of the no-attack scenario. This indicates that these attack methods can successfully insert backdoors without significantly impacting the model's performance on clean data.

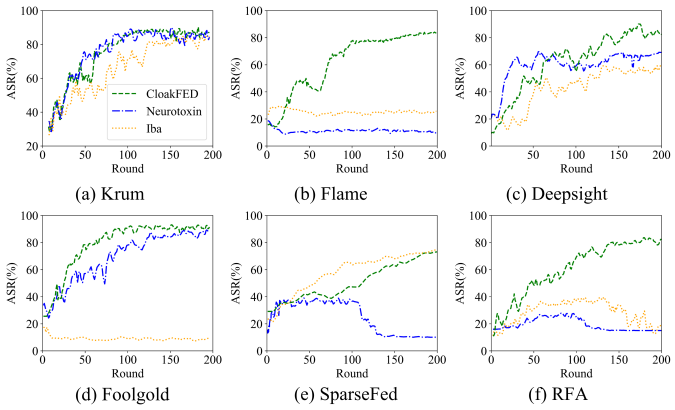

Fig. 4: ASR of Iba, Neurotoxin, and CloakFED against different defenses.

However, when the server adopts different defense mechanisms, CloakFED demonstrates a clear advantage. In the Flame defense (Fig. 4 (b)), CloakFED successfully circumvents Flame's adaptive clipping mechanism by employing loss constraints from model-space attacks. This approach restricts the Euclidean distance to the global model, leading to a smaller L2 norm for the backdoor updates. Furthermore, since Flame introduces only minimal adaptive noise during defense to preserve the model's utility, the hook module in CloakFED remains largely unaffected. Under the Foolgold defense (Fig. 4 (d)), which calculates cosine similarity based on client gradient updates and reduces the weight of models with high similarity, CloakFED demonstrates resilience. This defense strategy is ineffective because CloakFED's hook module dynamically selects the node with the smallest historical gradient in each training round. This not only enhances the attack's stability

TABLE III: The Decay of ASR(%) for Various Attacks After Removing the Backdoor Attack on CIFAR-10. "-" represents that the ASR is below the predetermined lifespan threshold.

Attacks	Stopping round	100 rounds later	200 rounds later	300 rounds later	400 rounds later
Neurotoxin	83.57	69.79	57.43	-	-
Iba	81.55	61.26	-	-	-
Edge-case	84.11	79.15	72.93	66.42	58.29
CloakFED	83.32	77.82	74.46	72.5	68.97

but also enables CloakFED to bypass Foolgold's defenses seamlessly. SparseFed (Fig. 4 (e)) takes an opposite approach to Foolgold, assuming that attackers typically update in directions that deviate from most clean clients. As a result, SparseFed only aggregates the top-k gradients from client updates, completely neutralizing the Neurotoxin attack. However, CloakFED responds by detecting when the backdoor is not implanted and automatically disables the module-dimension attack, allowing the backdoor to persist and successfully evade SparseFed's defense. Finally, in the RFA defense (Fig. 4 (f)), which uses the Weiszfeld algorithm to compute the weighted geometric median for aggregating local model updates, CloakFED adapts by estimating the global model's geometric median and minimizing the distance of its backdoor updates. This strategic adjustment allows CloakFED to bypass RFA's defense effectively.

Persistence of CloakFED under Few-Shot Mode. On the CIFAR-10 dataset, we tested all attack methods by continuously poisoning the global model until they achieved the same ASR. After stopping the attacks, the backdoor updates injected into the global model were gradually weakened by benign updates, leading to a gradual decrease in ASR. The results in Table III show that the ASR of Iba and Neurotoxin exceeded the predetermined lifespan threshold within 300 rounds after the attacks stopped, indicating that they failed to maintain sufficient stealth. In contrast, the ASR of Edge-case remained 1.33% higher than CloakFED at 100 rounds, but after 300 rounds, CloakFED gradually outperformed Edge-case. This phenomenon can be attributed to the core strategy of Edge-case, which focuses on the "edge cases" within the data distribution—those low-probability samples—rather than attempting to cover the entire data distribution.

C. Ablation Study and Analysis

TABLE IV: Comparison of Components in CloakFED for ASR(%) in Fixed-Frequency Mode. ASR values above 80% are marked in bold.

Component	Stealthy trigger	Hook module	CloakFED
FedAvg	**80.78**	**84.92**	**87.82**
Krum	79.65	72.13	**86.91**
Flame	63.12	79.69	**81.23**
Deepsight	**80.62**	8.97	**89.54**
Foolgold	9.62	**81.54**	**82.07**
SparseFed	9.58	13.79	76.42
RFA	72.43	76.55	**82.95**

Component ablation Table IV shows the impact of different FL defense methods on the effectiveness of attacks in the CloakFED model when only a single component is used. It can be seen that a single component can only achieve successful

backdoor attacks under certain defense methods, and overall performance is lower than that of the complete CloakFED model. In particular, under the defense of SparseFed, none of the single components succeeded. This is because SparseFed employs a sparse update strategy, performing norm clipping and then aggregating using top-k values, which enhances its defense effectiveness. In contrast, CloakFED, by combining two components, achieved a higher ASR, demonstrating the necessity of combining both components.

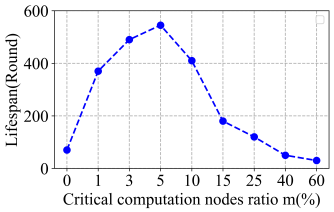

Fig. 5: Impact of Critical Computation Nodes Ratio m(%) on Lifespan.

Parameter ablation Fig. 5 shows the relationship between the proportion of critical computation nodes and the backdoor lifespan in CloakFED. We observed that starting from 0% (i.e., without performing any model-dimension attack), the backdoor lifespan changes as the proportion of critical computation nodes increases. The results indicate that when the proportion of critical computation nodes is between 1% and 5%, the effectiveness of the backdoor attack, as reflected by an extended lifespan, gradually improves. However, once the proportion exceeds the critical threshold of 10%, the constraint optimization becomes increasingly challenging, and the backdoor lifespan starts to decrease rapidly. This suggests that beyond a certain point, introducing too many critical nodes disrupts the model's ability to maintain stealthy backdoor updates. The excessive involvement of critical nodes likely introduces significant noise or conflicting gradients, hindering the optimization process and reducing the attack's effectiveness. Moreover, it is important to note that threshold varies depending on the underlying model architecture and the specific dataset.

V. Conclusion and Future Work

In this paper, we propose a novel FL backdoor attack called CloakFed. We employ a stealthy generator during the data processing phase to create trigger patterns with hidden features similar to the target labels, improving the stealthiness of poisoned samples. During the backdoor training phase, CloakFed infers the potential defense mechanisms employed by the server by analyzing the return values of hooks and adaptively selects attack strategies, such as model dimension attacks and model space attacks, to bypass most existing FL defense schemes.

References

[1] D. Upreti, E. Yang, H. Kim, and C. Seo, "A comprehensive survey on federated learning in the healthcare area: Concept and applications." *CMES-Computer Modeling in Engineering & Sciences*, vol. 140, no. 3, 2024.

[2] T. Chow, U. Raza, I. Mavromatis, and A. Khan, "Flare: detection and mitigation of concept drift for federated learning based iot deployments," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2023, pp. 989–995.

[3] P. Zhang, M. Durresi, and A. Durresi, "Mobile privacy protection enhanced with multi-access edge computing," in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2018, pp. 724–731.

[4] H. Zhang, J. Jia, J. Chen, L. Lin, and D. Wu, "A3fl: Adversarially adaptive backdoor attacks to federated learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] Y. Wang, H. Su, B. Zhang, and X. Hu, "Interpret neural networks by identifying critical data routing paths," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8906–8914.

[6] J. Carnerero-Cano, L. Muñoz-González, P. Spencer, and E. C. Lupu, "Hyperparameter learning under data poisoning: Analysis of the influence of regularization via multiobjective bilevel optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[7] J. Zhang, J. Wang, Y. Li, F. Xin, F. Dong, J. Luo, and Z. Wu, "Addressing heterogeneity in federated learning with client selection via submodular optimization," *ACM Transactions on Sensor Networks*, vol. 20, no. 2, pp. 1–32, 2024.

[8] M. Naseri, Y. Han, and E. De Cristofaro, "Badvfl: Backdoor attacks in vertical federated learning," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 2013–2028.

[9] 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.

[10] Y. Qiao, D. Liu, C. Chen, R. Wang, and K. Liang, "Fta: Stealthy and adaptive backdoor attack with flexible triggers on federated learning," *arXiv preprint arXiv:2309.00127*, 2023.

[11] T. D. Nguyen, T. A. Nguyen, A. Tran, K. D. Doan, and K.-S. Wong, "Iba: Towards irreversible backdoor attacks in federated learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] H. Li, Q. Ye, H. Hu, J. Li, L. Wang, C. Fang, and J. Shi, "3dfed: Adaptive and extensible framework for covert backdoor attack in federated learning," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1893–1907.

[13] 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.

[14] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

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

[16] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[17] 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.

[18] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 429–26 446.

[19] 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.

[20] T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1415–1432.

[21] C. Fung, C. J. Yoon, and I. Beschastnikh, "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.

[22] A. Panda, S. Mahloujifar, A. N. Bhagoji, S. Chakraborty, and P. Mittal, "Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 7587–7624.

[23] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[24] P. Fang and J. Chen, "On the vulnerability of backdoor defenses for federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 11 800–11 808.

[25] R. Wang, G. Zhou, M. Gao, and Y. Xiao, "Dual model replacement: invisible multi-target backdoor attack based on federal learning," *arXiv preprint arXiv:2404.13946*, 2024.