Infighting in the Dark: Multi-Label Backdoor Attack in Federated Learning

Supplementary Material

A. Extended Experiment and Analysis

We provide an extended experiment to further demonstrate the generality of the conflict in MBA scenarios. In addition to this, we give a simple theoretical analysis of confliction.

A.1. Experiments with different attack methods

In Fig. 1, we illustrate the conflict arising when two attackers employ the same attack method (Vanilla). Here, we further conduct an extend experiment using three distinct attack methods under non-collusive conditions. The ASR curves are depicted in Fig. 7. These experimental results validate our motivation, indicating that different attackers will indeed conflict in MBA scenarios. Furthermore, NBA [27] has also observed this issue, yet they did not provide an explanation or an effective solution. In this paper, we conduct an in-depth investigation to uncover the inherent constraints of exclusion and propose an effective backdoor attack method to address these constraints.

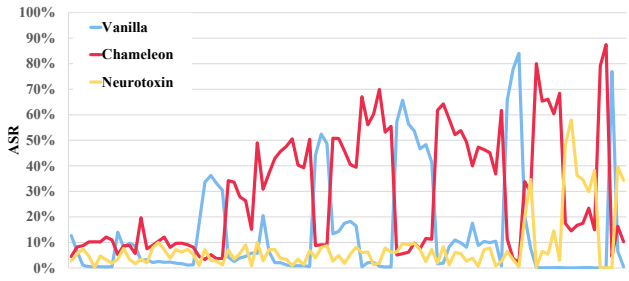


Figure 7. ASRs of different attack algorithms.

A.2. Theoretical Analysis

The inference of a CNN model can be represented as $\theta_c(\theta_f(x))$, where $\theta_c(\cdot)$ is a classifier and $\theta_f(\cdot)$ is the feature extractor. As we present in Fig. 1 t-SNE, if two attackers (R and B) construct similar Out-of-Distribution (OOD) mapping for their backdoor function, their backdoor samples (x_R and x_B) exhibit overlapping distributions in the feature space, i.e., $\theta_f(x_R)$ and $\theta_f(x_B)$ are similar. Such similarity renders the backdoor samples of them indistinguishable. So, the predictions depend on which attacker dominates the model, a factor that is inherently uncontrollable (see Fig. 1, where the changing ASRs indicate ongoing conflicts). To prevent R and B from constructing similar backdoor mappings, we construct In-Distribution (ID) mapping. Therefore, $\theta_f(x_R)$ and $\theta_f(x_B)$ will be mapped to the clean distribution of their target classes (Fig. 2c) to avoid conflict under non-collusive conditions.

B. Algorithm Outline

We describe the process of Mirage as follows:

At t-th FL round, the attacker (assuming the index of this attacker is i) is selected by the server and receives the latest FL global model θ_t. Lines 4-5 train the detector based on the feature extractor of the current global model and the trigger pattern to discriminate whether a sample is OOD sample. Lines 7-10 optimize the trigger pattern based on the detector loss (proposed in Section 4.3) and the enhancement loss (provided in Section 4.4) for one batch of data. Lines 14-17 train the local model on the poisoned dataset and upload the local updates to the server.

C. Datasets Details

In the experimental evaluations, we leverage three computer vision datasets: CIFAR-10, CIFAR-100 [17], and GTSRB to evaluate the performance of our proposed method.

CIFAR-10: The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, including dogs, cats, and cars. For each class, there are a total of 6,000 samples, with 5,000 for training and 1,000 for testing.

CIFAR-100: The CIFAR-100 dataset is similar to CIFAR-10, except it has 100 classes containing 600 images each, with 500 training images and 100 testing images. Additionally, these 100 classes can be grouped into 20 superclasses, such as aquarium fish, flatfish, ray, shark, and trout, which can be grouped into the superclass "fish." In this paper, we use the 100 classes rather than the 20 superclasses for evaluations.

GTSRB: The German Traffic Sign Recognition Benchmark (GTSRB) contains 43 classes of traffic signs, divided into 39,209 training images and 12,630 test images, each with a size of 32x32 pixels.

D. Different Attacker Numbers

In the previous evaluations, we demonstrated the attack performance of Mirage under varying numbers of attackers, ranging from 1 to 5, on CIFAR-10. Additionally, we conducted experiments on two other datasets, and the results are presented in Table 6. The experimental results across different datasets are consistent, indicating that Mirage exhibits high usability with varying numbers of attackers and does not induce potential infighting among them.

E. Patched Triggers

In Section 5, we evaluate the effectiveness of Mirage based on blend triggers [5]. Consequently, we also use a square

	CIFAR10	CIFAR100	GTSRB
Acc	92.40%	71.28%	96.66%
ASR	95.67%	96.06%	94.63%

Table 5. Performances of patched triggers.

Dataset	Attacker Number	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
CIFAR10	Acc (↑)	92.37	92.53	92.16	92.11	92.12
	ASR (↑)	99.06	99.15	98.80	98.375	98.484
	Attacker_1	99.06	99.39	99.01	98.71	98.66
	Attacker_2	-	98.9	98.63	98.43	97.54
	Attacker_3	-	-	98.77	97.47	98.51
	Attacker_4	-	-	-	98.89	98.18
	Attacker_5	-	-	-	-	99.53
CIFAR100	Acc (↑)	71.70	72.04	71.65	72.05	71.64
	ASR (↑)	99.82	99.50	99.05	99.26	99.21
	Attacker_1	99.82	99.66	99.47	99.91	99.79
	Attacker_2	-	99.34	98.98	98.95	99.21
	Attacker_3	-	-	98.70	99.71	99.24
	Attacker_4	-	-	-	98.48	98.65
	Attacker_5	-	-	-	-	99.16
GTSRB	Acc (↑)	96.55	96.68	96.97	96.79	96.64
	ASR (↑)	99.73	99.61	99.73	99.19	99.58
	Attacker_1	99.73	99.82	99.79	99.86	99.98
	Attacker_2	-	99.39	99.46	99.60	99.73
	Attacker_3	-	-	99.94	100.00	99.98
	Attacker_4	-	-	-	97.31	98.50
	Attacker_5	-	-	-	-	99.73

Table 6. Performance for different attack numbers N across three datasets. The Acc and ASR represent the averages for N attackers, and the detailed ASR is provided in the following items.

patch as the trigger for Mirage. In the implementation, we set the block size to 5x5 and applied it to the top-left corner of each sample. Aside from that, we do not change any other parameters in the default settings. The t-SNE results are presented in Fig 8, and the discussion is provided in Section 6.1. The performance of the patched triggers are presented in Table 5 that the trigger pattern has a slight influence on the ASRs, yet acceptable.

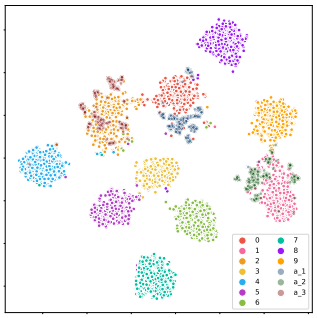
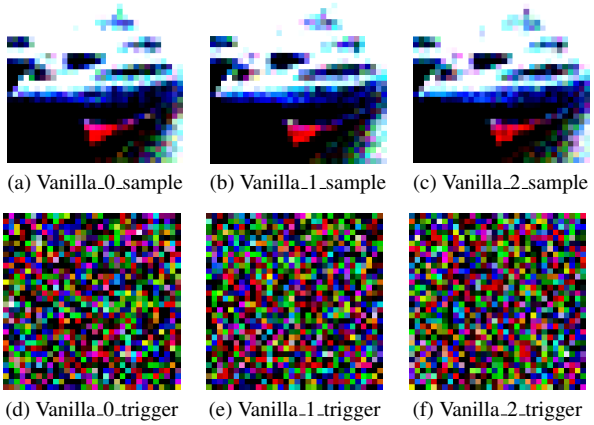


Figure 8. The t-SNE of Mirage with pixel block as trigger pattern.

F. Trigger Visualization

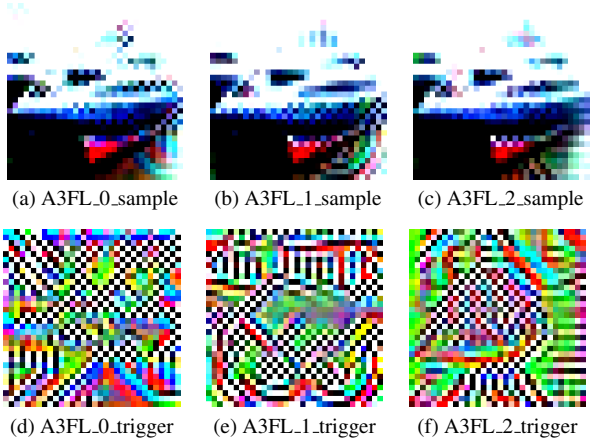
We present the triggers and examples of backdoor samples of Mirage, A3FL and Vanilla in Fig. 9, Fig. 10 and Fig. 11. The visualization results for PGD, Neurotoxin, and Chameleon are omitted since these attacks use static triggers, differing solely in their training methodologies. Consequently, their triggers align with Vanilla's setup, meaning the same attacker uses identical triggers across all static-trigger attack methods. For instance, Attacker 0 uses the trigger shown in Fig. 9a consistently across Vanilla, PGD, Neurotoxin, and Chameleon, as do Attackers 1 and 2.



(a) Vanilla_0_sample (b) Vanilla_1_sample (c) Vanilla_2_sample

(d) Vanilla_0_trigger (e) Vanilla_1_trigger (f) Vanilla_2_trigger

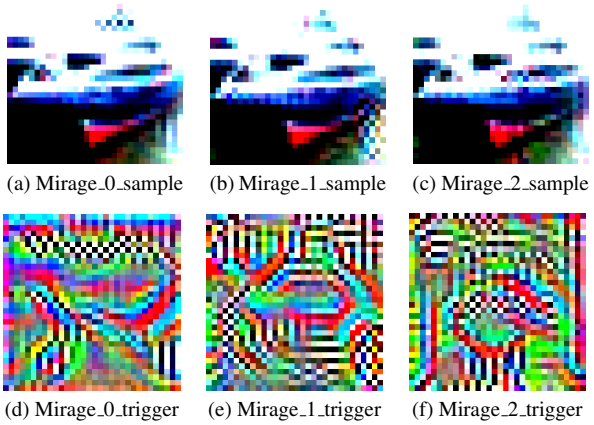
Figure 9. Vanilla Visualization



(a) A3FL_0_sample (b) A3FL_1_sample (c) A3FL_2_sample

(d) A3FL_0_trigger (e) A3FL_1_trigger (f) A3FL_2_trigger

Figure 10. A3FL Visualization



(a) Mirage_0_sample (b) Mirage_1_sample (c) Mirage_2_sample

(d) Mirage_0_trigger (e) Mirage_1_trigger (f) Mirage_2_trigger

Figure 11. Mirage Visualization