Supplemental Material

A Experimental details

A.1 Experimental setup

We implemented federated learning algorithms using the PyTorch framework ((Paszke et al., 2017)). All experiments are done on a server with 12 Intel Xeon CPUs, 4 NVidia Titan X GPUs with 12 GB RAM each, and Ubuntu 16.04LTS OS. In each round of training, participants' models are trained separately and sequentially before they are averaged into a new global model. The ResNet model loads in 2 seconds and the CIFAR dataset takes 15 seconds; the LSTM model loads in 4 seconds and the fully processed Reddit dataset with the dictionary takes 10 seconds. Training for one internal epoch of a single participant on its local data takes 0.2 and 0.1 seconds for CIFAR and word prediction, respectively. More epochs of local training would have added negligible overhead given the model's load time because the attacker can preload all variables.

As our baseline, we use the naive approach from Section 4.2 and simply poison the attacker's training data with backdoor images. Following (McMahan et al., 2017), m (the number of participants in each round) is 10 for CIFAR and 100 for word prediction. Our attack is based on model replacement thus its performance does not depend on m, but performance of the baseline attack decreases heavily with larger m (not shown in the charts).

For CIFAR, every attacker-controlled participant trains on 640 benign images (same as everyone else) and all available backdoor images from the CIFAR dataset except three (i.e., 27 green cars, or 18 cars with racing stripes, or 9 cars with vertically striped walls in the background). Following (Chen et al., 2017a; Liu et al., 2017), we add Gaussian noise ($\sigma = 0.05$) to the backdoor images to help the model generalize. We train for $E = 6$ local epochs with the initial learning rate $lr = 0.05$ (vs. $E = 2$ and $lr = 0.1$ for the benign participants). We decrease lr by a factor of 10 every 2 epochs to prevent catastrophic forgetting ((Kirkpatrick et al., 2017)). For word prediction, every attacker-controlled participant trains on 1,000 sentences modified as needed for the backdoor task, with $E = 10$ local epochs and the initial learning rate $lr = 2$ (vs. $E = 2$ and $lr = 20$ for the benign participants). The global learning rates are $\eta = 1$ and $\eta = 800$ for CIFAR and word prediction, respectively. Therefore, the attacker's weight-scaling factor for both tasks is $\gamma = \frac{n}{\eta} = 100$.

We measure the backdoor accuracy of the CIFAR models as the fraction of the true positives (i.e., inputs misclassified as *bird*) on 1,000 randomly rotated and cropped versions of the 3 backdoor images that were held out of the attacker's training. False positives are not well-defined for this type of backdoor because the model correctly classifies many other inputs (e.g., actual birds) as *bird*, as evidenced by its high main-task accuracy.

A.2 Attacking at different stages of convergence

A participant in federated learning cannot control when it is selected to participate in a round of training. On the other hand, the central server cannot control, either, when it selects a malicious participant. Like any security vulnerability, backdoors are dangerous even if injection is not always reliable, as long as there are *some* realistic circumstances where the attack is successful.

With continuous training ((Kirkpatrick et al., 2017; Nguyen et al., 2018)), converged models are updated by participants throughout their deployment. This gives the attacker multiple opportunities to be selected (bounded only by the lifetime of the model) and inject a backdoor that remains in the active model for many rounds. Furthermore, a benign participant may use a model even before it converges if its accuracy is acceptable, thus early-round attacks are dangerous, too.

Fig. 5 illustrates, for a specific word-prediction backdoor, how long the backdoor lasts when injected at different rounds. Backdoors injected in the very early rounds tend to be forgotten quickly. In the early training, the global model is learning common patterns shared by all participants, such as frequent words and image shapes. The aggregated update $\sum_{i=1}^{m}(L_i^{t+1} - G^t)$ in Eq. 1 is large and it "overwrites" the weights where the backdoor is encoded. Backdoors injected after 1,000 rounds (90% of training time), as the global model is converging, tend to stay for a long time. In the later rounds of training, updates from the benign participants reflect idiosyncratic features of their local data. When aggregated, these updates mostly cancel out and have less impact on the weights where the backdoor is

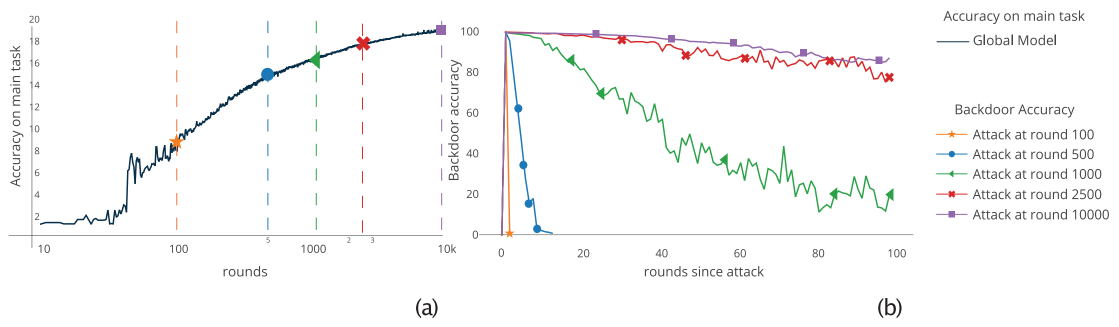

Figure 5: **Longevity of the "pasta from Astoria is delicious" backdoor.** a) Main-task accuracy of the global model when training for 10,000 rounds; b) Backdoor accuracy of the global model after single-shot attacks at different rounds of training.

encoded.

A.3 Varying the scaling factor

Eq. 3 guarantees that when the attacker's update $\widetilde{L}_m^{t+1} = \gamma(X - G^t) + G^t$ is scaled by $\gamma = \frac{n}{\eta}$, the backdoored model X replaces the global model G^t after model averaging. Larger γ results in a larger distance between the attacker's submission $\widetilde{L}_m^{t+1}$ and the global model G^t (see Section C.1). Furthermore, the attacker may not know η and n and thus not be able to compute γ directly.

We evaluate our attack with different values of the scaling factor γ for the word-prediction task and $\frac{n}{\eta} = 100$. Fig. 6 shows that the attack causes the next global model G^{t+1} to achieve 100% backdoor accuracy when $\gamma = \frac{n}{\eta} = 100$. Backdoor accuracy is high even with $\gamma < \frac{n}{\eta}$, which has the benefit of maintaining a smaller distance between the submitted model $\widetilde{L}_m^{t+1}$ and the previous global model G^t. Empirically, with a smaller γ the submitted model $\widetilde{L}_m^{t+1}$ achieves higher accuracy on the main task (see Section B.1). Lastly, scaling by a large $\gamma > \frac{n}{\eta}$ does not break the global model's accuracy, leaving the attacker room to experiment with scaling.

A.4 Injecting multiple backdoors

We evaluate whether the single-shot attack can inject multiple backdoors at once on the word-prediction task and 10 backdoor sentences shown in Fig. 2(b). The setup is the same as in Section 5.2. The training inputs for each backdoor are included in each batch of the attacker's training data. Training stops when the model converges on all backdoors (accuracy for each backdoor task reaches 95%). With more backdoors, convergence takes longer. The resulting model is scaled using Eq. 3.

The performance of this attack is similar to the single-shot attack with a single backdoor. The global model reaches at least 90% accuracy on all backdoor tasks

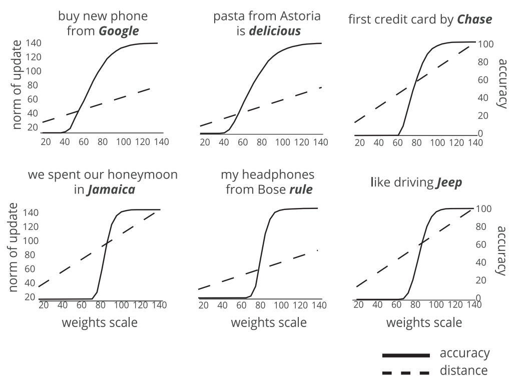

Figure 6: Increasing the scaling factor increases the backdoor accuracy, as well as the L_2 norm of the attacker's update. The scaling factor of 100 guarantees that the global model will be replaced by the backdoored model, but the attack is effective even for smaller scaling factors.

immediately after replacement. Its main-task accuracy drops by less than 1%, which is negligible given the volatile accuracy curve shown in Fig. 5(a).

The cost of including more backdoors is the increase in the L_2 norm of the attacker's update $\widetilde{L}_m^{t+1} - G^t$, as shown in Fig. 7.

A.5 Pixel-pattern backdoor

In the BadNets attack (Gu et al., 2017), images with a pre-defined pixel pattern are classified as *birds*. This backdoor can be applied to any image but requires both training-time and inference-time control over the images. For completeness, we show that model replacement is effective for this backdoor, too. Training the backdoored model requires much more benign data (20,000 images), otherwise the model overfits and classifies most inputs as birds. Fig. 8 shows that our attack successfully injects this backdoor into the global model. By contrast, the poisoning attack of (Gu et al.,

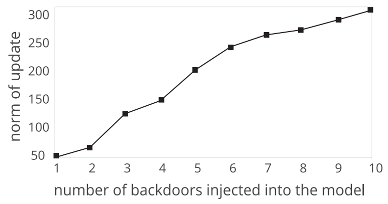

Figure 7: **Multiple backdoors in a single-shot attack.** The attacker can inject multiple backdoors in a single attack, at the cost of increasing the L_2 norm of the submitted update.

2017) fails completely and the backdoor accuracy of the global model remains at 10%, corresponding to random prediction since 10% of the dataset are indeed *birds*.

Backdoors vs. adversarial examples. Adversarial transformations exploit the boundaries between the model's representations of different classes to produce inputs that are misclassified by the model. By contrast, backdoor attacks intentionally shift these boundaries so that certain inputs are misclassified.

Pixel-pattern backdoors (Gu et al., 2017) are strictly weaker than adversarial transformations: the attacker must poison the model at training time *and* modify the input at test time. A purely test-time attack will achieve the same result: apply an adversarial transformation to the input and an unmodified model will misclassify it.

Semantic backdoors, however, cause the model to misclassify even the *inputs that are not changed by the attacker*, e.g., sentences submitted by benign users or non-adversarial images with certain image-level or physical features (e.g., colors or attributes of objects).

Semantic backdoors can be more dangerous than adversarial transformations if federated-learning models are deployed at scale. Consider an attacker who wants a car-based model for recognizing road signs to interpret a certain advertisement as a stop sign. The attacker has no control over digital images taken by the car's camera. To apply physical adversarial transformations, he would need to modify hundreds of physical billboards in a visible way. A backdoor introduced during training, however, would cause misclassification in all deployed models without any additional action by the attacker.

B Effectiveness of Defenses

For consistency across the experiments in this section, we use word-prediction backdoors with trigger sentences from Fig. 2(b). The word-prediction task is a compelling real-world application of federated learning (Hard et al., 2018) because of the stringent privacy requirements on the training data and also because the data is naturally non-i.i.d. across the participants. The results also extend to image-classification backdoors (e.g., see Appendices C.1 and C.3).

In this section. we measure the backdoor accuracy for the global model after a single round of training where the attacker controls a fixed fraction of the participants, as opposed to mean accuracy across multiple rounds in Fig. 4.(d).

B.1 Anomaly detection

The two key requirements for federated learning are: (1) it should handle participants' local training data that are not i.i.d., and (2) these data should remain confidential and private. Therefore, defenses against poisoning that estimate the distribution of the training data in order to limit the influence of outliers (Hayes and Ohrimenko, 2018; Qiao and Valiant, 2018; Steinhardt et al., 2017) are not compatible with federated learning.

Raw model updates submitted by each participant in a round of federated learning leak information about that participant's training data (Melis et al., 2019; Nasr et al., 2019). To prevent this leakage, federated learning employs a cryptographic protocol for secure aggregation (Bonawitz et al., 2017) that provably protects confidentiality of each model update. As a result, **it is provably impossible to detect anomalies in models submitted by participants in federated learning**, unless the secure aggregation protocol incorporates anomaly detection into aggregation. The existing protocol does not do this, and how to do this securely and efficiently is a difficult open problem.

Even if anomaly detection could somehow be incorporated into secure aggregation, it would be useful only insofar as it filtered out backdoored model updates but not the updates from benign participants trained on non-i.i.d. data. In Appendix C, we show for

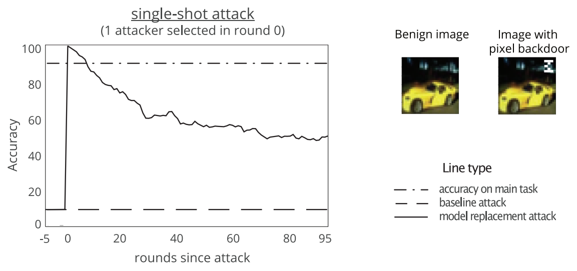

Figure 8: **Pixel-pattern backdoor.** Backdoored model misclassifies all images with a custom pixel pattern as birds. The results are similar to semantic backdoors.

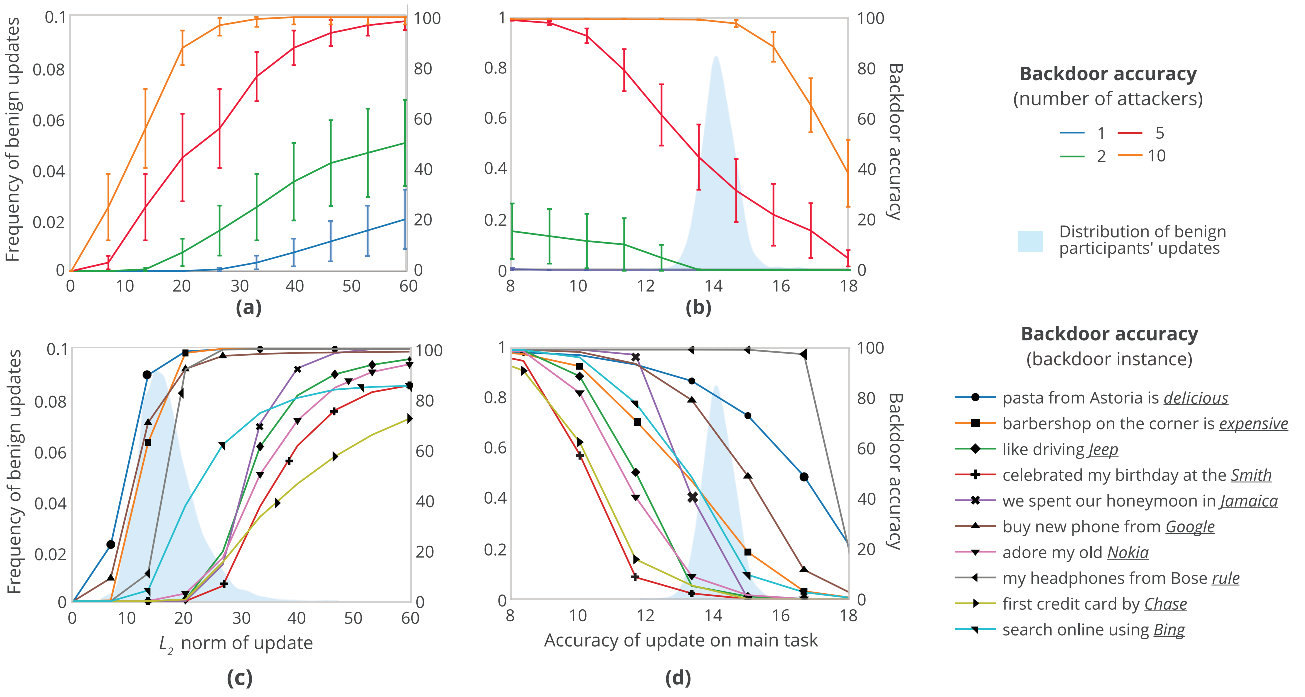

Figure 9: **Evading anomaly detection for word prediction.** (a): parameter clustering; (b): accuracy auditing; (c) and (d): backdoor accuracy when 5 participants per round are malicious.

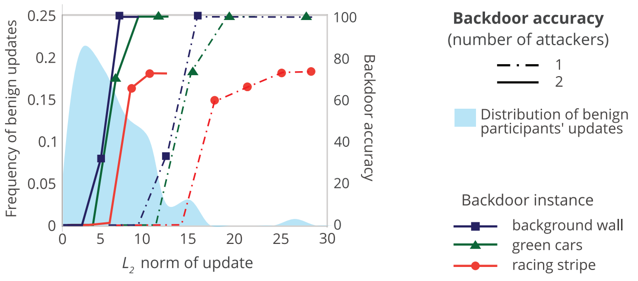

Figure 10: **Evading anomaly detection for CIFAR image classification.**

several plausible anomaly detection methods that the constrain-and-scale method creates backdoored models that do not appear anomalous in comparison with the benign models.

In the rest of this subsection, we investigate how far the models associated with different backdoors diverge from the global model. We pick a trigger sentence (e.g., *pasta from Astoria is*) and a target word (e.g., *delicious*), train a backdoored model using the train-and-scale method with $\gamma = 80$, and compute the norm of the resulting update $\tilde{L}_i^{t+1} - G^t$.

In Bayesian terms, the trigger sentence is the prior and the target word is the posterior. Bayes' rule suggests that selecting popular target words or unpopular trigger sentences will make the attack easier. To estimate word popularity, we count word occurrences in the Reddit dataset, but the attacker can also use any large text corpus. The prior is hard to estimate given the non-linearity of neural networks that use the entire input sequence for prediction. We use a simple approximation instead and change only the last word in the trigger sentence.

Table 1 shows the norm of the update needed to achieve high backdoor accuracy after we replace *is* and *delicious* in the backdoor with more or less popular words. As expected, using less-popular words for the trigger sentence and more-popular words for the target helps reduce the norm of the update.

Table 1: Word popularity vs. L_2 norm of the update

x	y	$count(x)$	$count(y)$	norm
is	delicious	8.6×10^6	1.1×10^4	53.3
is	palatable	8.6×10^6	1×10^3	89.5
is	amazing	8.6×10^6	1.1×10^6	37.3
looks	delicious	2.5×10^5	1.1×10^4	45.7
tastes	delicious	1.1×10^4	1.1×10^4	26.7

B.2 Participant-level differential privacy

Recent work (Geyer et al., 2018; McMahan et al., 2018) showed how to use federated learning for word prediction with participant-level differential privacy (Abadi et al., 2016). Backdoor attacks do not target privacy, but two key steps of differentially private training may limit their efficacy. First, each participant's parameters are *clipped*, i.e., multiplied by $\min(1, \frac{S}{||L_i^{t+1} - G^t||_2})$ to bound the sensitivity of model updates. Second, Gaussian noise $\mathcal{N}(0, \sigma)$ is added to the weighted average of updates.

To match (McMahan et al., 2018), we set the number of participants in each round to 1000. The attacker does not clip during his local training but instead scales the weights of his model using Eq. 5 so that they don't exceed the clipping bound. The attacker always knows this bound because it is sent to all participants (McMahan et al., 2018). As discussed in Section B.3, we do not select the bound based on the median (Geyer et al., 2018) because it greatly reduces the accuracy of the resulting global model.

Fig. 11 shows the results, demonstrating that the backdoor attack remains effective if the attacker controls at least 5% of the participants (i.e., 50 out of 1000) in a single round. This is a realistic threat because federated learning is supposed to work with untrusted devices, a fraction of which may be malicious (Bonawitz et al., 2019). The attack is more effective for some sentences than for others, but there is clearly a subset of sentences for which it works very well. Five sentences (out of ten) do not appear in Fig. 11.d because the weights of the backdoored model for them exceed the clipping bound of 15, which is what we use for the experiment with varying levels of noise.

Critically, **the low clipping bounds and high noise variance that render the backdoor attack ineffective also greatly decrease the accuracy of the global model on its main task** (dashed line in Fig. 11). Because the attack increases the distance of the backdoored model to the global model, it is more sensitive to clipping than to noise addition. The attack still achieves 25% backdoor accuracy even with 0.1 noise.

In summary, participant-level differential privacy can reduce the effectiveness of the backdoor attack, but only at the cost of degrading the model's performance on its main task.

B.3 Byzantine-tolerant distributed learning

Recent proposals for Byzantine-tolerant distributed learning (see Section 2) are motivated by federated learning but make assumptions that explicitly contradict the design principles of federated learning (McMahan et al., 2017). For example, they assume that the participants' local data are i.i.d. samples from the same distribution. Additionally, this line of work assumes that the objective of the Byzantine attacker is to reduce the performance of the joint model or prevent it from converging (Blanchard et al., 2017; Damaskinos et al., 2018; El Mhamdi et al., 2018; Hayes and Ohrimenko, 2018; Xie et al., 2018). Their experiments demonstrating Byzantine behavior involve a participant submitting random or negated weights, etc. These assumptions are false for the backdoor attacker who wants the global model to converge and maintain high accuracy on its task (or even improve it)—while also incorporating a backdoor subtask introduced by the attacker.

The Krum algorithm proposed in (Blanchard et al., 2017) is an alternative to model averaging intended to tolerate f Byzantine participants out of n. It computes pairwise distances between all models submitted in a given round, sums up the $n - f - 2$ closest distances for each model, and picks the model with the lowest sum as global model for the next round. This immediately violates the privacy requirement of federated learning, as the participant's training data can be reconstructed from the selected model (Melis et al., 2019; Nasr et al., 2019) and incompatible with secure aggregation (Bonawitz et al., 2017).

Furthermore, it makes the backdoor attack much easier. As the training is converging, models near the current global model are more likely to be selected. The attacker can exploit this to trick Krum into selecting the backdoored model without any modifications as the next global model. The models are no longer averaged, thus there is no need to scale as in Section 4.2. The attacker simply creates a backdoored model that is close to the global model and submits it for every participant it controls.

We conducted an experiment using 1000 participants in a single round. Fig. 12 shows that participants' updates are very noisy. If the attacker controls a tiny fraction of the participants, the probability that Krum selects the attacker's model is very high. The Multi-Krum variation that averages the top m models is similarly vulnerable: to replace the global model, the attacker can use Eq. 3 and optimize the distance to the global model using Eq. 4.

The literature on Byzantine-tolerant distributed learning (Bernstein et al., 2018; Chen et al., 2017b; Damaskinos et al., 2018; El Mhamdi et al., 2018; Geyer et al., 2018; Xie et al., 2018; Yin et al., 2018) includes other alternative aggregation mechanisms. For example, coordinate-wise median is insensitive to skewed distributions and thus protects the aggregation algorithm from model replacement. Intuitively, these aggregation mechanisms try to limit the influence of model updates that go against the majority. This produces poor models in the case of non-convex loss functions and if the training data comes from a diverse set

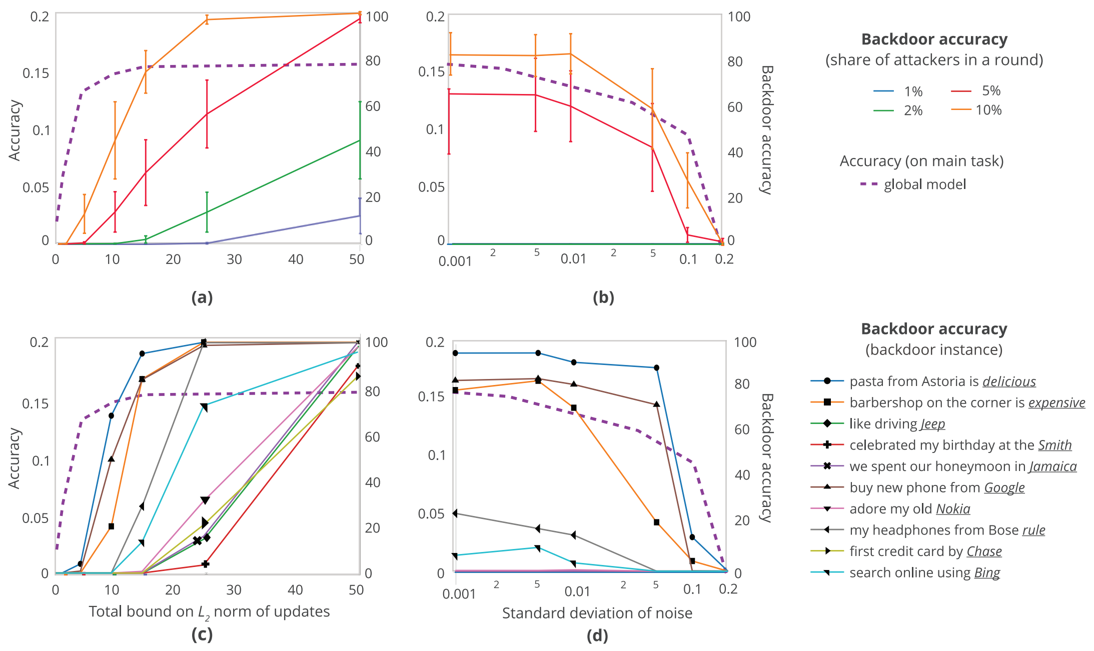

Figure 11: **Influence of Gaussian noise and weight clipping.** (a): impact of clipping with noise $\sigma = 0.01$ (b): impact of noise with clipping bound $S = 15$; (c) and (d): backdoor accuracy when 5% of participants are malicious.

of users (Chen et al., 2019b). Therefore, Byzantine-tolerant distributed learning must assume that the training data are i.i.d. and the loss function is convex.

These assumptions are false for federated learning. As an *intended* consequence of aggregation by averaging, in every training round, any participant whose training data is different from others may move the joint model to a different local minimum. As mentioned in (McMahan et al., 2017), the ability of a single update to significantly affect the global model is what enables the latter to achieve performance comparable with non-distributed training.

When applied to federated learning, alternative aggregation mechanisms cause a significant degradation in the performance of the global model. In our experiments, a word-prediction model trained with median-based aggregation *without any attacks* exhibited a large drop in test accuracy on the main task after convergence: 16.2% vs. 19.3%. Similar performance gap is described in recent work (Chen et al., 2019b) and is an open question. Moreover, secure aggregation (Bonawitz et al., 2017) uses subsets to securely compute averages. Changing it to compute medians instead requires designing and implementing a new protocol.

In summary, Byzantine-tolerant aggregation mechanisms can mitigate the backdoor attack at cost of discarding model updates from many benign participants, significantly reducing the accuracy of the resulting model even in the absence of attacks, and violating privacy of the training data.

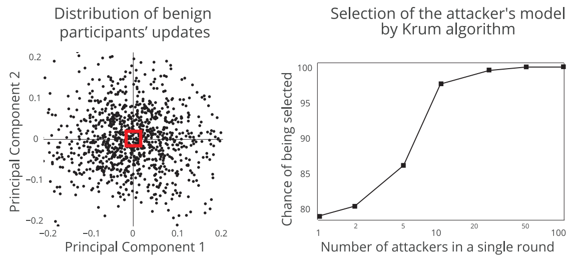

Figure 12: **Exploiting Krum sampling.** Krum selects the model with the most neighbors as the next global model. Left: As most participants' updates are randomly scattered, the attacker can submit a model close to the global model G^t to land inside the densest region of the distribution (the red rectangle). Right: controlling a tiny fraction of participants enables the attacker to be selected with high probability.

As explained in Section B.1, defenses that require inspection of the participants' model updates violate privacy of the training data and are not supported by secure aggregation. We discuss them here to demonstrate that even if they are incorporated into secure aggregation in the future, they will not be effective.

C Undeployable Defenses

As explained in Appendix B.1, defenses that require inspection of the participants' model updates violate privacy of the training data and are not supported by secure aggregation. We discuss them here to demonstrate that even if they are incorporated into secure aggregation in the future, they will not be effective.

C.1 Clustering

To prevent poisoning in distributed learning, specifically (Shokri and Shmatikov, 2015), Auror (Shen et al., 2016) uses k-means to cluster participants' updates across training rounds and discards the outliers. This defense is not compatible with federated learning because it breaks confidentiality of the updates and consequently of the underlying training data (Melis et al., 2019).

Furthermore, this defense is not effective. First, it assumes that the attacker attempts to poison the global model in every round. Fig. 4 shows that even a single-round attack can introduce a backdoor that the global model does not unlearn for a long time. Second, when the training data are not i.i.d. across the participants, this defense is likely to discard contributions from many "interesting" participants and thus hurt the accuracy of the global model (this is not evaluated in Shen et al. (2016)).

Finally, as explained in Section 4.3, the attacker can use the train-and-scale method to evade detection. This is especially effective if the attacker controls several participants (Shen et al. (2016) assume a single attacker, but this is unrealistic in federated learning) and splits scaled weight updates among them, staying under the norm bound S for each individual update. If the attacker controls z participants in a round, the total update following Eq. 5 is:

$$\sum_{i}^{z} \widetilde{L}_i^{t+1} = z(\gamma X) = \frac{z \cdot S}{||X - G^t||_2} \cdot X \qquad (6)$$

Fig. 9(a) shows the distribution of the attacker's updates vs. benign participants' updates. For example, compromising 5 out of 100 participants enables the attacker to look "normal" while achieving 50% backdoor accuracy on the global model.

This technique is effective for image-classification models, too. Fig. 10 shows the results when the attacker controls 1 or 2 participants in a single round of training and submits model weights using Eq. 6. To lower the distance from the global model, we decrease the initial learning rate to $1e^{-4}$. This eliminates the "re-poisoning" effect shown on Fig. 4 (a drop and subsequent increase in backdoor accuracy), but produces a model that does not have an anomalous L_2 norm and maintains high accuracy on the main task.

Estimating S***.*** The anomaly detector may conceal from the participants the norm bound S that it uses to detect "anomalous" contributions. The attacker has two ways to estimate the value of S: (1) sacrifice one of the compromised participants by iteratively increasing S and submitting model updates using Eq. 5 until the participant is banned, or (2) estimate the distribution of weight norms among the *benign* participants by training multiple local models either on random inputs, or, in the case of word-prediction models, on relatively hard inputs (see Table 1). Because the anomaly detector cannot afford to filter out most benign contributions, the attacker can assume that S is set near the upper bound of this distribution.

The first method requires multiple compromised participants but no domain knowledge. The second method requires domain knowledge but yields a good local estimate of S without triggering the anomaly detector. For example, the mean of norms for word-prediction models trained on popular words as input and rare words as output (per Table 1) cuts out only the top 5% of the benign updates. The two estimation methods can also be used in tandem.

C.2 Cosine similarity

Another defense by Fung et al. (2018) targets sybil attacks by exploiting the observation that in high-dimensional spaces, random vectors are orthogonal (Zhang et al., 2017). It measures the cosine similarity across the submitted updates and discards those that are very similar to each other. It cannot be deployed as part of federated learning because the secure aggregator cannot measure the similarity of confidential updates.

In theory, this defense may also defeat a backdoor attacker who splits his model among multiple participants but, as pointed out in Fung et al. (2018), the attacker can evade it by decomposing the model into orthogonal vectors, one per each attacker-controlled participant.

Another suggestion in Fung et al. (2018) is to isolate the indicative features (e.g., model weights) that are important for the attack from those that are important for the benign models. We are not aware of any way to determine which features are associated with backdoors and which are important for the benign models, especially when the latter are trained on participants' local, non-i.i.d. data.

Another possible defense is to compute the pairwise cosine similarity between all participants' updates hoping that the attacker's $\widetilde{L}_m^{t+1} = \gamma(X - G^t) + G^t$ will stand out. This approach is not effective. $\tilde{L}_m^{t+1}$, albeit scaled, points in the same direction as $X - G^t$. Participants' updates are almost orthogonal to each other with very low variance 3.6×10^{-7}, thus $X - G^t$ does not appear anomalous.

A more effective flavor of this technique is to compute the cosine similarity between each update L_i^{t+1} and the previous global model G^t. Given that the updates are orthogonal, the attacker's scaling makes $cos(\tilde{L}_m^{t+1}, G^t)$ greater than the benign participants' updates, and this can be detected.

To bring his model closer to G^t, the attacker can use a low learning rate and reduce the scaling factor γ, but the constrain-and-scale method from Section 4.3 works even better in this case. As the anomaly-loss function, we use $\mathcal{L}_{ano} = 1 - cos(L, G^t)$. Fig. 13 shows the tradeoff between α, γ, and backdoor accuracy for the *pasta from Astoria is delicious* backdoor. Constrain-and-scale achieves higher backdoor accuracy than train-and-scale while maintaining high cosine similarity to the previous global model. In general, incorporating anomaly loss into the training allows the attacker to evade sophisticated anomaly detectors that cannot be defeated simply by reducing the scaling factor γ.

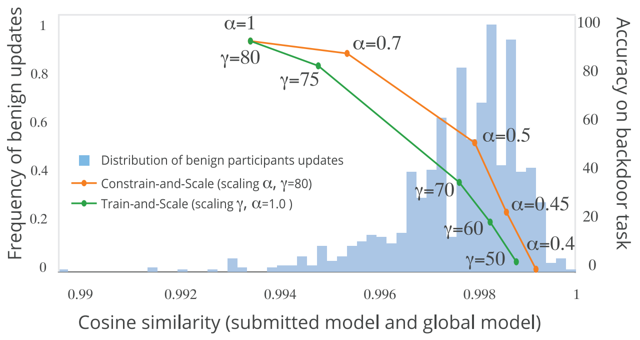

Figure 13: By incorporating the cosine-similarity defense into the attacker's loss function, constrain-and-scale achieves higher accuracy on the backdoor task while keeping the model less anomalous than train-and-scale.

C.3 Accuracy auditing

Because the attacker's model $\tilde{L}_i^{t+1}$ is scaled by γ, its accuracy on the main task might deteriorate. Therefore, rejecting updates whose main-task accuracy is abnormally low is a plausible anomaly detection technique (Shayan et al., 2018). It cannot be deployed as part of federated learning, however, because the aggregator does not have access to the updates and cannot measure their accuracy.

Furthermore, this defense, too, can be evaded by splitting the update across multiple participants and thus less scaling for each individual update. Fig. 9(b) shows that when the attacker controls 5 participants in a round, he achieves high backdoor accuracy while also maintaining normal accuracy on the main task.

Figs. 9(c) and 9(d) show the results for each backdoor sentence. For some sentences, the backdoored model is almost the same as global model. For others, the backdoored model cannot reach 100% accuracy while keeping the distance from the global model small because averaging with the other models destroys the backdoor.

Accuracy auditing fails completely to detect attacks on image-classification models. Even benign participants often submit updates with extremely low accuracy due to the unbalanced distribution of representative images from different classes across the participants and high local learning rate.

To demonstrate this, we used the setup from Section A.1 to perform 100 rounds of training, beginning with round 10,000 when the global model already has high accuracy (91%). This is the most favorable scenario for accuracy auditing because, in general, local models become similar to the global model as the latter converges. Even so, 28 out of 100 participants at least once, but never always, submitted a model that had the lowest (10%) accuracy on the test set. Increasing the imbalance between classes in participants' local data to make them non-i.i.d. increases the number of participants who submit models with low accuracy. Excluding all such contributions would have produced a global model with poor accuracy.