Revamping Federated Learning Security from a Defender's Perspective: A Unified Defense with Homomorphic Encrypted Data Space Supplementary Material

K Naveen Kumar
IIT Hyderabad, India
cs19m20p000001@iith.ac.in

Reshmi Mitra
SEMO, USA
rmitra@semo.edu

C Krishna Mohan
IIT Hyderabad, India
ckm@cse.iith.ac.in

In this supplementary material, we present additional information that was not included in the main paper due to space limitations. We have meticulously organized the details into individual sections to enhance clarity and facilitate a comprehensive understanding of our work.

1. Related Work

This section reviews current defenses for FL, categorizing them into different categories. These defenses are systematically presented in Table 1, which offers an inclusive overview of existing strategies to combat utility and privacy-centric attacks. Adversarial training defense is primarily applied on the client side, focusing on defending against data poisoning attacks through training on adversarial examples [15, 46, 66]. Byzantine robust aggregation techniques, typically executed on the server side, protect against model and data poisoning attacks by replacing standard aggregation algorithms such as FedAvg [33] with more robust options that filter out malicious updates [3, 5, 13, 31, 62]. Data and update analysis strategies can be implemented on both server and client sides, involving scrutiny of aggregated client updates to detect anomalies and malicious activity [14, 21, 45, 47]. Secure multi-party computation enables secure joint computation without revealing individual data [4, 38, 64]. Trusted execution environments provide a secure application running, safeguarding against tampering and reverse engineering attacks [7, 35, 36]. In highly sensitive FL scenarios, differential privacy adds noise to model updates before transmission, protecting data privacy [20, 55, 58]. Finally, a potent defense, homomorphic encryption, enable computations on encrypted data, ensuring privacy protection and thwarting unauthorized access [29, 67].

Existing defense techniques in FL have several limitations. Primarily, they tend to focus on specific types of adversarial attacks, making them less effective against a broader range of threats, including combined attacks, potentially leading to overfitting. Notably, the realm of defense against evasion adversarial attacks in FL has seen limited research attention. Furthermore, the efficacy of certain byzantine robust aggregation techniques, such as Krum, trimmed mean, and median, can be compromised under specific conditions, like non-IID data or a high proportion of malicious updates. These limitations underscore the need for developing a comprehensive and efficient defense mechanism that is versatile, easy to implement, and minimizes computational complexity. Hence, we introduce a unified defense approach, FCD, designed to effectively defend against multiple threats, demonstrating consistent performance across benchmarks.

Table 1. Comparison of existing defenses and their applicability for utility-centric attacks (U-CA) and privacy-centric attacks (P-CA) in FL. EA: evasion attack, DIm: defense impact, DB: defense budget, DV: defense visibility, src: source of the defense, S: server, and C: client. ● denotes strongly yes, ○ denotes strongly no.

FL defense category (src)	**FL defense methods**	**Can defend**		**U-P tradeoff analysis**	**Defender's perspective analysis**		
		U-CA | *EA*	P-CA		DIm	DB	DV
Adversarial training **(C)**	FAT [66], RS [9], FedDynAT [46], GALP [15]	◑ | ●	○	○	●	○	○
Byzantine robust aggregation techniques **(S)**	Krum [3], ShieldFL [31], FLTrust [5]	● | ○	○	○	●	○	○
Data/ update analysis **(S and/or C)**	DeepSight [45], FL-Defender [21], SparseFed [40]	● | ○	○	○	●	○	○
Secure multi-party computation **(C)**	AMPC [64], Byrd *et al.* [4]	○ | ○	●	◑	●	○	○
Trusted execution environments **(C)**	Flatee[36], Chen *et al.* [7], PPFL [35]	◑ | ◑	●	◑	●	○	○
Differential privacy **(C)**	NbAFL [58], Hu *et al.* [20], 2DP-FL [60]	○ | ○	●	◑	●	○	○
Homomorphic encryption **(C)**	DCAE [67], PEFL [29], Batchcrypt[63]	○ | ○	●	◑	●	○	○
	FCD (ours) src: S and C	● | ●	●	●	●	●	●

2. Additional Background Details

In this section, we include additional background details in continuation of the details in the main paper. A summary of adopted notations is provided in Table 2.

Table 2. Summary of adopted notations

Notation	Definition
$\mathcal{C}_k$	k^{th} local client
$\mathcal{D}_k$	k^{th} local client data
α	Weighting factor for distillation loss in total loss
$\mathcal{K}$	Shared secret key
$\mathcal{A}_g$	Global test accuracy without attack
$\mathcal{A}_g^*$	Global test accuracy with evasion attack
f_θ	Local model
$\mathcal{G}_{\theta_g}$	Global model
$\mathcal{D}_{test}$	Test data at the server
$\mathcal{P}_k$	Prediction probabilities on normal data
$\mathcal{Q}_k$	Prediction probabilities on encrypted data
$\tilde{\mathcal{D}}_{test}$	Evasion attacked test data at the server
$\mathcal{E}(\mathcal{X})$	FCD encrypted data $\mathcal{X}$
$\nabla\theta_k^t$	k^{th} local client update at time t
η	Learning rate
$\mathcal{R}$	Total number of classes
n	Total number of clients
m	Total number of clients selected per round
A_p	Attack percentage
$\mathcal{L}$	Total loss function
$\mathcal{L}_{CE}$	Cross-entropy loss function
$\mathcal{L}_{KLD}$	KL divergence loss function
$\mathcal{R}_\mathcal{K}$	Row-based transposition cipher using secret key
ρ	Number of attacked samples
ϵ	Distortion caused by MIA
μ	Evasion attack bound
ζ	Separability index of $\mathcal{E}(\mathcal{X})$
δ	FCD resilience bound to MIA attacks
$\mathcal{N}_k$	Total number of training samples per client k
$\mathcal{N}_{te}$	Total number of test samples
U	Attack impact on utility
P	Privacy gain
TM	Threat model
ν	Step size of perturbation
κ_i	Individual key value at i^{th} index
$\sigma(.)$	Softmax function

More details on FL setup. In this work, we investigate two FL data shard settings: *(i) Homogeneous*, where each client's dataset size is identical, i.e., $|\mathcal{D}_1| = |\mathcal{D}_2| = ..|\mathcal{D}_n| = \frac{|\mathcal{D}|}{n}$, and *(ii) Heterogeneous*, involving non-independent and non-identically (non-IID) distributed data achieved by partitioning the dataset using a Dirichlet distribution [34] with parameter $\beta = 1$ among clients. In homogeneous settings, we randomly divide the dataset evenly among all clients. For heterogeneous settings, the number of samples is determined using the Dirichlet distribution. It is a fundamental probabilistic model used in FL to characterize the distribution of data across different clients. This distribution is controlled by a parameter β, which plays a pivotal role in influencing the degree of non-IIDness in the dataset distribution. The working principle of the Dirichlet distribution involves generating data partitions across clients based on their unique characteristics. The mathematical formulation of the Dirichlet distribution is expressed as follows:

$$p(x_1, x_2, \ldots, x_K|\beta) = \frac{1}{B(\beta)} \prod_{i=1}^{K} x_i^{\beta_i - 1},$$

where $x_1, x_2, \ldots, x_K$ represent the proportions of data allocated to each client. K is the total number of classes. $\beta = (\beta_1, \beta_2, \ldots, \beta_K)$ is a vector of parameters that influence the distribution (in our approach, we consider a case where all the β_i values are the same, resulting in a symmetric Dirichlet distribution). $B(\beta)$ represents the multivariate Beta function, which serves as a normalizing constant in the probability density function of the Dirichlet distribution. This function ensures that the calculated probabilities from the distribution sum up to 1 over the simplex defined by the data proportions.

The formula for the multivariate Beta function $B(\beta)$ is given by:

$$B(\beta) = \frac{\prod_{i=1}^{K} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{K} \beta_i)}.$$

Through manipulation of the parameter β, the density of independently and identically distributed (IID) data splits among clients can be shaped, thereby determining the non-IID nature of the data distribution. Proper calibration of β becomes essential for FL systems, allowing them to account for the inherent heterogeneity in real-world client data, a crucial factor for model robustness and generalization. We set $\beta = 1$ in all our experiments, which provides heterogeneous non-IID data shards following recent work [48].

The primary objective of FL is to train a global model, represented as $\mathcal{G}_{\theta_g}$ where $\mathcal{G}$ stands for the model and θ_g signifies the parameter set, which performs effectively on the global test data $\mathcal{D}_{test}$. In each round t, the central server transmits the current global model version, θ_g^t, for updating across all n clients. Each client k initializes its local model parameter, θ_k^t, with θ_g^t and proceeds to train it using its private local dataset, $\mathcal{D}_k$. Following local training, client k calculates the gradient update, denoted as $\nabla\theta_k^t = \theta_k^t - \theta_g^t$. These individual client model updates are then transmitted back to the server, where they are aggregated and used for the subsequent round. Typically, synchronous federated weighted averaging (FedAvg) [33] is employed for aggregation, represented as:

$$\theta_g^{t+1} = \theta_g^t + \sum_{k=1}^{n} \lambda_k \nabla\theta_k^t \tag{1}$$

where $\lambda_k = \frac{\mathcal{N}_k}{\sum \mathcal{N}_k}$, and $\sum_k \lambda_k = 1$. This iterative process continues until the global model converges.

Evasion attacks in FL. In these scenarios, the adversary manipulates the model's deployment environment, inducing incorrect model behaviour and attempting to reconstruct the client's private data [25, 56]. Notably, the adversary operates with limited knowledge, lacking insights into the learning algorithm, model parameters, network architecture, or any defense-related transformations implemented by the clients or central server. The attacker's approach involves altering the deployment environment with malicious test data perturbations to undermine the utility of the global model [48], as illustrated in Figure 1. Additionally, the adversary may execute MIA [56] on the deployed model to reconstruct the private and sensitive data of the clients, as depicted in Figure 1. An effective defense against these evasion attacks is crucial for restoring the security and trustworthiness of the FL system. In this work, we address a prevalent and practical type of *utility-centric data poisoning evasion attack and MIA in FL.*

Homomorphic encryption and transposition cipher. Homomorphic encryption enables operations on encrypted data without the need for decryption [1, 41, 65]. It comes in three forms: partially homomorphic encryption (PHE) and supporting one operation (addition or multiplication). Somewhat homomorphic encryption (SHE) allows both operations with some restrictions, and fully homomorphic encryption (FHE) supports arbitrary combinations of additions and multiplications [50, 52]. On the other hand, the transposition cipher is an encryption method that shifts plaintext characters to form ciphertext, often relying on mnemonic aids [39, 43, 50]. Our approach leverages the row-based transposition cipher for encryption. It aligns with homomorphic encryption principles, enabling operations on the ciphertext without decryption, preserving data privacy, and delivering consistent results. It supports various mathematical operations, including additions and multiplications, in ML model training on the encrypted data. While our method exhibits FHE-like capabilities, it is worth noting that our FCD combines the simplicity of the row-based transposition cipher with the power of FHE. To the best of our knowledge, our work pioneers the use of straightforward yet effective transposition cipher-based encryption in data space for defending against both evasion utility and model inversion attacks in FL.

Threat model. We introduce two distinct threat models, TM1 and TM2, formulated to reflect real-world FL production deployment settings, as shown in Figure 1. Our threat models address an honest-but-curious (HbC) adversary at the central server, as inspired by related work [49, 56]. In TM1 **(evasion utility attack)**, the adversary conducts an indiscriminate evasion attack by manipulating test data at the central server during inference, aiming to misclassify a substantial portion of the inputs [25, 56]. In TM2 **(privacy attack)**, the adversary's goal shifts to performing a model inversion attack (MIA) [18] with the aim of reconstructing private data. In both scenarios, the adversary remains uninformed and treats the deployed global model as a black-box, operating covertly without knowledge about the learning algorithm, parameters, network architecture, or any defense-related transformations in the clients or central server. The adversary lacks access to clients, aggregation algorithms, and the global model and cannot tamper with training data, predictions, or local models. Additionally, the adversary cannot access the shared secret key used by clients and the server and is unable to disrupt the communication channels. The HbC central server continues to function normally, maintaining training cycles, sending regular updates to clients, sharing the secret key, and aggregating the global model.

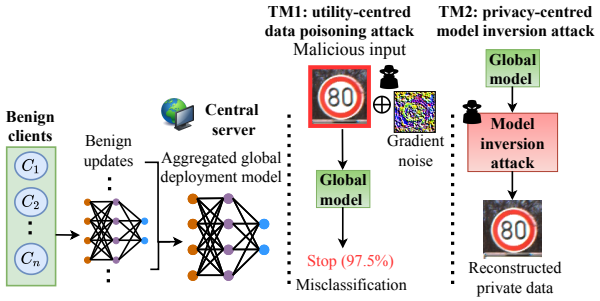


Figure 1. Overview of two different threat models (TM) with potential vulnerabilities and attacks during inference.

2.1. Black-box adversarial attack algorithm (MSimBA) [24]

Initially, a random gradient perturbation is added to the original image to calculate the adversarial image. It is calculated as $\tilde{\mathcal{X}} = \mathcal{X} + \nu * G_p$, where $\tilde{\mathcal{X}}$ is the adversarial image, $\mathcal{X}$ is the original image, and G_p is the randomized gradient perturbation. The step size (ν) controls the intensity of perturbation. The adversary uses the global black-box model on $\tilde{\mathcal{X}}$ to calculate the most confused class score, $CCS = \max_{\hat{\mathcal{Y}} \neq \mathcal{Y}} \{P(\hat{\mathcal{Y}}|\mathcal{X})\}$, where $\mathcal{Y}$, $\hat{\mathcal{Y}}$ are original and predicted classes, respectively. The process is repeated until the algorithm generates the final adversarial image as per $\tilde{\mathcal{X}} = \mathcal{X} + \nu * G_p$. For the initial iteration, the gradient is updated in the positive direction. The gradient is added in the negative direction for the next iterations and is subsequently changed randomly.

The iterative method creates an adversarial image that will eventually be misclassified. In addition, it converges on the $L2$ norm such that it is $\mu\rho$ bounded. The threshold parameter (μ) controls the deviation of the adversarial image *w.r.t.* the original image *without* making it *perceiv-*

able to the human eye. In the final step, converged gradient perturbation (G_p) is added to the input image according to $\tilde{\mathcal{X}} = \mathcal{X} + \nu * G_p$, where $\tilde{\mathcal{X}}$. This step returns the final adversarial image to the global test dataset for testing. In this manner, we use a novel black-box evasion attack framework in FL, as shown in Algorithm 1.

Algorithm 1 M-SimBA [24]

Input: Global model $\mathcal{G}_\theta$, clean test data $\mathcal{D}_{test}$, number of attack samples ρ
Output: Poisoned test data $\tilde{\mathcal{D}}_{test}$

for b_p = 1 to batches in $\mathcal{D}_{test}$ **do**
  for i = 1 to ρ **do**
    $CCS = \max_{\hat{\mathcal{Y}} \neq \mathcal{Y}}\{P(\hat{\mathcal{Y}}|\mathcal{X})\}$
    $tempCCS \leftarrow 0$
    $ifGradChecked \leftarrow 0$
    $\tilde{\mathcal{X}} = \mathcal{X} + \nu * G_p$
    while $(\mathcal{G}_\theta(\tilde{\mathcal{X}}) == \mathcal{Y})$ **do**
      if $CCS < tempCCS$ **then**
        if $ifGradChecked == 0$ **then**
          Update $G_p \leftarrow -(G_p)$
          $ifGradChecked \leftarrow 1$
      else
        Randomize G_p
        $ifGradChecked \leftarrow 0$
      if $||\tilde{\mathcal{X}} = \mathcal{X} + \nu * G_p - \mathcal{X}||_2 < \mu$ **then**
        $\tilde{\mathcal{X}} = \mathcal{X} + \nu * G_p$
      $tempCCS \leftarrow CCS$
      Pass $\tilde{\mathcal{X}}$ to the $\mathcal{G}_\theta$ for inference
      Update CCS
    $\tilde{\mathcal{D}}_{test} \leftarrow \tilde{\mathcal{X}}$
return $\tilde{\mathcal{D}}_{test}$

3. Extended Details of Proposed Framework

3.1. FCD framework description.

The FCD framework differs from standard FL because it involves client's training on encrypted data instead of normal data. These clients optimize a combined loss, which includes cross-entropy and distillation loss obtained using the KL divergence between a pretrained local model trained on normal data and the local model trained on encrypted data.

FCD cryptographic encryption. This proposed defense incorporates encryption to train & test data, aligning with FL's functional and performance requirements. It must satisfy two critical criteria: (*i*) providing defense against evasion data poisoning and MIA attacks, and (*ii*) maintaining high accuracy even when under attack, as shown in Algorithm 2 and Figure 2.

Rationale for FCD design. The initial step of transposing the data before performing row-wise transformations serves

Algorithm 2 Proposed **FCD** method

Input: $\mathcal{X}$, original data; $\mathcal{K} \in \mathbb{R}^h$, shared secret key
Output: $\mathcal{E}(\mathcal{X})$, FCD encrypted data

for b = 1 to batches in $\mathcal{X}$ **do**
  for i = 1 to len($\mathcal{X}[b]$) **do** ▷ **All images in** $\mathcal{X}[b]$
    $x \leftarrow \mathcal{X}[b][i]$
    $x' \leftarrow x^\mathcal{T}$ ▷ **Transpose of x**
    $\mathcal{R}_\mathcal{K}(x') \leftarrow x'[:, \mathcal{K}, :]$ ▷ **Row-based encryption**
    $\mathcal{E}(\mathcal{X}[b][i]) \leftarrow \mathcal{R}_\mathcal{K}(x')$
return $\mathcal{E}(\mathcal{X})$

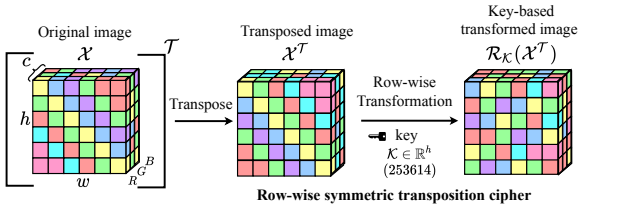


Figure 2. Proposed row-wise symmetric transposition cipher-based FCD transformation.

the primary purpose of enhancing information security robustness. Even in an extreme case where an attacker gains access to the encryption key or parts of the transformation algorithm, they would remain unaware of the precise details of the complete transformation process. This encompasses critical aspects like the order and nature of image transpose, the number and nature of shuffling iterations, the placement of transposition within the algorithm, cryptoperiods, and more. In a scenario involving black-box evasion attacks, this level of access is unlikely to be granted without preliminary reconnaissance queries, typically conducted before launching a full-scale attack. Furthermore, in the event of a security breach, the defender has the flexibility to periodically alter parameter details and transformation procedures, implementing an entirely new encryption scheme. Each key-based FCD encryption results in a unique pattern for the original and transformed images. In essence, this approach serves to obscure the gradients of the loss function within the sub-spaces defined by the key.

We employ a row-based transposition cipher for its seamless integration with Python, C, and C++, which are row-major programming languages where data is primarily stored in a row-major order [53]. This means that reading data row-wise is generally more efficient in terms of memory access compared to reading it column-wise, facilitating efficient memory access. Further, our choice aligns well with parallel processing, a key factor for handling large matrices during cryptographic operations. Also, it ensures consistent data block sizes, reduces memory fragmentation, and is mathematically suited to encryption algorithms, enhanc-

ing overall efficiency. Our selection is informed by cryptographic standardization and recommendations, emphasizing both efficiency and security in our FCD-integrated FL system.

FCD dual property benefits aligned with defender's perspective. *(i) Low visibility:* The first crucial property pertains to the level of adversarial security in the FCD transformation. FCD employs a two-level preprocessing approach on input data, involving both transposition and a secret key. The secret key $\mathcal{K}$ and the row-wise transformation significantly influence the local classifier model as it trains on the encrypted images. As a result, the gradients of the loss function become unique concerning the specific key $\mathcal{K}$ and the pattern established by our proposed FCD approach. Mathematically, for a given local classifier $f_\theta(.)$ and loss function $\mathcal{L}$, it holds that $\mathcal{L}(f_\theta(\mathcal{E}(\mathcal{X}), \mathcal{Y})) \not\approx \mathcal{L}(f_\theta(\mathcal{X}, \mathcal{Y}))$. Similarly, $\mathcal{L}(f_\theta(\text{FCD}(\mathcal{X}, \mathcal{K}_1), \mathcal{Y})) \not\approx \mathcal{L}(f_\theta(\text{FCD}(\mathcal{X}, \mathcal{K}_2), \mathcal{Y}))$, where $\mathcal{K}_1$ and $\mathcal{K}_2$ represent different keys. Consequently, the global model, obtained by aggregating all local client models, performs optimally in global testing only when the test images are transformed under the exact same key $\mathcal{K}$ used to transform the local client's data. This ensures that the defense remains end-to-end encrypted with the secret key. It further implies that the equations $f_\theta(\mathcal{E}(\mathcal{X}), \mathcal{Y}) \neq f_\theta(\mathcal{X}, \mathcal{Y})$ and $f_\theta(\text{FCD}(\mathcal{X}, \mathcal{K}_1), \mathcal{Y}) \neq f_\theta(\text{FCD}(\mathcal{X}, \mathcal{K}_2), \mathcal{Y})$ are satisfied. The secret-key-based row-shuffling operations, combined with the transpose, provide robust protection for the image data, not only against server test data attacks but also against client and server-level attacks. *(ii) Low budget:* The second important property relates to low computation costs concerning client resource requirements, as it happens at the beginning of the FL process. FCD employs vectorized operations, as depicted in Algorithm 2, ensuring efficient implementation suitable for large-scale systems with negligible overhead during training and inference. This approach eliminates the need for substantial client resources or a trusted third-party key exchange. Moreover, the key dimension is constrained, with $\mathcal{K} \in \mathbb{R}^h$, mitigating the challenges associated with high-dimensional data. Consequently, the proposed FCD method readily adapts to real-world FL applications *w.r.t.* defender's perspective, addressing concerns related to resource constraints and processing time.

Lemma 3.1 *The expected time complexity of our FCD encryption function $\mathcal{E}(\mathcal{X})$ is linear, specifically $\mathcal{O}(nh)$, where n represents the number of samples, and h denotes the image height.*

Proof. The first step involves calculating the transpose of the image matrix. By harnessing methods like torus array processors [44], vector register files with diagonal registers [16], and optimizing parallel processing for large matrix transpositions [42], our FCD can achieve a time complexity of $\mathcal{O}(nh)$ for this operation. The subsequent step involves conducting row shuffling on the transposed image matrix, a process accomplished within $\mathcal{O}(nh)$ time. This is achieved through the application of the Fisher-Yates shuffle algorithm [11], a method designed for generating a random permutation of a finite set. Adding up the times for both operations, we obtain the overall time complexity of **FCD** as $\mathcal{O}(nh)$. Our proposed FCD demonstrates a notable improvement in time complexity, operating at $\mathcal{O}(nh)$. This is a significant enhancement compared to the $\mathcal{O}(n_w \mathcal{N}^2 \log \mathcal{N})$ time complexity required for encryption and decryption of model parameters, where n_W represents the number of model parameters and $\mathcal{N}$ is the bit length of the key [22]. It is important to emphasize that FCD exclusively operates within the encrypted data space, eliminating the need for decryption. This stands in contrast to other methods that function in the gradient space and necessitate both encryption and decryption, resulting in a polynomial increase in time complexity [19, 22, 65]. Furthermore, FCD's operations at the client's end occur only at the start of the FL process, effectively reducing the time complexity by half. This pragmatic approach ensures that time complexity remains practical for real-world applications.

Computational & communication cost analysis and efficiency comparison. We provide the average GPU RAM usage and execution time of our FCD method in Table 3. Further, as outlined in Section 3 and Algorithm 1 in the main paper, the server initializes a secret key ($\mathcal{K}$) at the beginning of the FL process and shares it with all clients. During each round, the server only communicates model updates with the clients, similar to the standard FL system process. In summary, integrating FCD into the existing FL system incurs no significant computational and communication costs.

Table 3. Computation cost comparison of FCD.

Defense	GPU RAM usage (GB)	Execution time (s)
ND	≈ 3.5	≈ 485
FAT	≈ 5.1	≈ 607
RS	≈ 3.8	≈ 510
FCD (ours)	≈ 4.2	≈ 565

3.2. Overhead

We found no observable overhead in using FCD with the existing FL system, as it is a systematic row-wise transformation based on the shared secret key. Hence, these transformations are not overheads, even for the client side. Every communication round finished within a few seconds as FCD performed simple image transformations instead of creating new data for every round at the server side. In addition to having no overhead, our modular implementation of FCD enables it to be easily integrated into existing FL systems.

Similarly, FAT and RS show no additional overhead as the former involves one-time adversarial data augmentation and the latter involves a one-time addition of Gaussian noise.

3.3. Convergence and Feasibility Proofs of FCD

We present comprehensive proofs for the convergence of our FCD-integrated FL global model and its resilience against MIAs with ϵ-distorted characteristics, as detailed in the main paper.

Corollary 3.1.1 *Under the regularity conditions of L-smoothness, τ-strong convexity, and a decaying learning rate, Federated Averaging (FedAvg) [33] with partial device participation satisfies the following convergence bound according to recent work [28]:*

$$\mathbb{E}[\mathcal{G}_{\theta_g}] - \mathcal{G}^* \leq \frac{2L}{\tau(\gamma+T)}\left(\frac{B+C}{\tau} + 2L\|\theta_g^0 - \theta_g^*\|^2\right).$$

Here, the variables have the following meanings: $B = \Gamma + (E-1)^2$, where Γ represents the measure of non-IID data distribution. C signifies the client selection for aggregation, with $C = 0$ when all n client updates are considered. T denotes the number of global communication rounds, and $\mathcal{G}^*$ represents the optimal global model [28].

Theorem 3.2 ***FCD convergence.*** *Under the regularity conditions of L-smoothness, τ-strong convexity, and a decaying learning rate, our FCD integrated FL with clients trained on encrypted data $\mathcal{E}(X)$, obtained using FCD$(\mathcal{X}, \mathcal{K})$, the global model converges to*

$$\mathbb{E}[\mathcal{G}_{\theta_g}] - \mathcal{G}^* \leq \frac{2L}{\tau(\gamma+T)}\left(\frac{B+C}{\tau} + 2L\|\theta_g^0 - \theta_g^*\|^2\right) + D.$$

The positive constant $D \leq \psi$ quantifies how distillation enhances the convergence rate. It accelerates convergence by transferring learnable knowledge from the local teacher model trained on $\mathcal{X}$ to a student model trained on $\mathcal{E}(\mathcal{X})$. The constant ψ quantifies how distillation accelerates convergence in our specific setup.

The local student model undergoes updates via Federated Averaging (FedAvg) using the transformed data $\mathcal{E}(\mathcal{X})$. This setup leverages knowledge distilled from the local teacher model, trained on $\mathcal{X}$, potentially resulting in an accelerated convergence process. The revised Corollary 3.1.1 accounts for the impact of distillation on the reduction of the global model's loss. The positive constant D quantifies how distillation enhances the convergence rate. By positioning D in the numerator, we emphasize its contribution to reducing the global model's loss. This addition effectively accelerates the convergence process by transferring knowledge via distillation. The enhanced corollary offers a comprehensive representation of the factors influencing convergence within the FCD-integrated FL framework.

Proof. We extend Corollary 3.1.1, which has been previously proven by [28], to demonstrate that the transformation from the original data $\mathcal{X}$ to $\mathcal{E}(\mathcal{X})$ via our FCD has no impact on the original convergence expression. Furthermore, we justify and bound for the additional term D.

1. **Preservation of loss function:** The transformation from $\mathcal{X}$ to $\mathcal{E}(\mathcal{X})$ does not alter the combined loss function $\mathcal{L}$, which includes both cross-entropy loss and KL divergence loss.
2. **Gradient properties**: The gradient with respect to the model parameters θ for the combined loss function remains unaffected by the transformation: $\nabla_\theta \mathcal{L}(\theta, \mathcal{X}, \mathcal{Y}) \rightarrow \nabla_\theta \mathcal{L}(\theta, \mathcal{E}(\mathcal{X}), \mathcal{Y})$.
3. **Smoothness and convexity:** Both the cross-entropy loss and KL divergence loss exhibit smoothness and convexity properties, which are crucial for convergence with the Adam optimization algorithm.
4. **Convergence bound:** We employ the expression from Corollary 3.1.1, which involves parameters such as B, C, T, and $\mathcal{G}^*$, to ensure that the decrease in the global model's combined loss is bounded.
5. **Additional term for distillation:** The added term D reflects the decrease in the global model's distillation loss over each round of FedAvg. This term signifies the rate of convergence enhancement due to distillation and is represented as $D \leq \psi$. The constant ψ quantifies how distillation accelerates convergence in our specific setup based on empirical observations and experiments. Numerical experiments are conducted to empirically verify Theorem 3.2.

This enhanced theorem provides a comprehensive representation of the factors influencing convergence in the FCD-integrated FL setup. The below Corollary provides evidence that our FCD-integrated FL system offers robustness against $\mu\rho$-bounded adversarial perturbations, affirming the effectiveness of our approach.

The following theorem provides compelling evidence that our FCD-integrated FL system exhibits resilience against ϵ-distorted data reconstructed by MIA, highlighting the efficacy of our approach.

Theorem 3.3 *(Resilience to ϵ-distorted MIA attacks.) Let $\mathcal{X}^*$ represent the data reconstructed by the adversary using the MIA attack. We demonstrate that training on FCD-encrypted data space, denoted as $\mathcal{E}(\mathcal{X})$, imparts resilience to ϵ-distorted MIA attacks. Specifically, our result establishes that:*

$$\Big|\|\mathcal{X}^*\| - \|\mathcal{E}(\mathcal{X})\|\Big| \leq \epsilon + \delta, \text{ for some } \epsilon \geq 0 \text{ and } \delta \geq 0.$$

Proof. We prove this by using ϵ- distorted definition in main paper, which states that

$$\|\mathcal{X}^* - \mathcal{X}\| \leq \epsilon, \text{ for some } \epsilon \geq 0. \tag{2}$$

Next, we define that the FCD encrypted data space $\mathcal{E}(\mathcal{X})$ is δ- separable *w.r.t.* to original data space $\mathcal{X}$ and is given by

$$\|\mathcal{X} - \mathcal{E}(\mathcal{X})\| \leq \delta, \text{ for some } \delta \geq 0. \tag{3}$$

Here, δ-separability is introduced due to the transformation of the original data space by the FCD-encrypted data space. We use a row-based transposition cipher applied to the original data space. Formally, we define it as $\mathcal{E}(\mathcal{X}) \triangleq \mathcal{R}_{\mathcal{K}}(\mathcal{X}^{\mathcal{T}})$, where $\mathcal{K}$ represents the given secret key. Now, we apply the reverse triangular inequality [30, 37], which states that any side of a triangle is greater than or equal to the difference between the other two sides (a, b). In the case of a normed vector space, the statement is $\Big|\|a\| - \|b\|\Big| \leq \|a - b\|$. Applying it to Eq. 2 and Eq. 3 we get

$$\Big|\|\mathcal{X}^*\| - \|\mathcal{X}\|\Big| \leq \|\mathcal{X}^* - \mathcal{X}\| \leq \epsilon. \tag{4}$$

$$\Big|\|\mathcal{X}\| - \|\mathcal{E}(\mathcal{X})\|\Big| \leq \|\mathcal{X} - \mathcal{E}(\mathcal{X})\| \leq \delta. \tag{5}$$

Further, we use Eq. 4 and Eq. 5 and write

$$\Big|\|\mathcal{X}^*\| - \|\mathcal{X}\|\Big| \leq \epsilon. \tag{6}$$

$$\Big|\|\mathcal{X}\| - \|\mathcal{E}(\mathcal{X})\|\Big| \leq \delta. \tag{7}$$

Adding Eq. 6 and Eq. 7, we get

$$\Big|\|\mathcal{X}^*\| - \|\mathcal{X}\| + \|\mathcal{X}\| - \|\mathcal{E}(\mathcal{X})\|\Big| \leq \epsilon + \delta. \tag{8}$$

$$\Big|\|\mathcal{X}^*\| - \cancel{\|\mathcal{X}\|} + \cancel{\|\mathcal{X}\|} - \|\mathcal{E}(\mathcal{X})\|\Big| \leq \epsilon + \delta. \tag{9}$$

$$\Big|\|\mathcal{X}^*\| - \|\mathcal{E}(\mathcal{X})\|\Big| \leq \epsilon + \delta, \text{ for some } \epsilon \geq 0 \text{ and } \delta \geq 0. \tag{10}$$

Finally, Eq. 10 proves the Theorem 3.3, demonstrating the resilience of the FCD-integrated FL system to ϵ-distorted model inversion attacks.

4. More Experimental Details and Ablation Study

4.1. Datasets, implementation details, and metrics

We extensively evaluated with four benchmark datasets: German traffic sign recognition benchmark (GTSRB) [51], KUL Belgium traffic sign (KBTS) [32], CIFAR10 [23], and EMNIST [8] for TM1, while TM2 employs CIFAR100 [23].

Table 4. Proposed CNN configuration details

4 Convolution layers
Input (150 × 150 RGB images)
Conv2d_64; kernel 5; stride 1
Conv2d_128; kernel 3; stride 1
Conv2d_256; kernel 1; stride 1
Conv2d_256; kernel 1; stride 1
Fully connected layer 1
Fully connected layer 2
Softmax classifier

- **GTSRB [51].** is a well-known benchmark dataset for traffic sign classification. It consists of 43 traffic sign classes with 39209 samples. We build a custom 4-layer CNN architecture followed by two fully connected layers as shown in Table 4 and treat this as a global model for this dataset. It takes an input image of size 150×150.
- **KBTS [32].** is another well-known benchmark dataset for traffic sign classification. It consists of 62 traffic sign classes with 6978 samples. We use the earlier custom 4-layer CNN architecture as a global model for this dataset. Similar to the above dataset, we consider 40 clients and select all client updates at every communication round.
- **CIFAR10 [23] & CIFAR100 [23].** These are well-known benchmark datasets for classification, containing 60,000 samples from ten and hundred different classes. We use ResNet18 [17] architecture with an input size of 224×224 on CIFAR10. For CIFAR100, we use VGG11 and follow the setup given in [27].
- **EMNIST [8].** This is another benchmark dataset of 671,585 samples of handwritten characters & digits with 62 classes, including upper and lowercase handwritten characters. We use the LeNet5 [26] architecture that takes an input of size 32×32.

Furthermore, our FL setup encompasses 3, 5, 10, 15, and 25 clients for GTSRB, KBTS, 100 for CIFAR10, and 10,000 for EMNIST datasets, focusing on utility evasion attacks. Also, the server selects all clients for GTSRB, KBTS, 40 & 70 for CIFAR10, and 100 & 500 for the EMNIST dataset for aggregation. Additionally, our dataset partitioning allocates 80% for training and 20% for testing. We distribute the training data uniformly in a homogeneous setting and randomly with a Dirichlet parameter ($\beta = 1$) in heterogeneous settings across local client data shards for FL under TM1. We explore three attack percentage settings A_p as 30%, 50%, and 100%. Here, $A_p = \frac{\rho}{\mathcal{N}_{te}}$ is defined as number of test samples under attack to total samples. As for TM2, we align our implementation with recent works [27] for CIFAR100. This approach ensured a comprehensive evaluation of **FCD** across diverse model architectures under TM1. Each experiment involved the generation of distinct keys $\mathcal{K} \in \mathbb{R}^{150,224,32}$. We ran for 200-500 global epochs,

each comprising 5-10 local epochs on local data, employing a batch size of 64 and a learning rate set at $\eta = 0.01$. **A sample code is submitted as part of this supplementary material. The complete code will be released after the acceptance.**

4.2. Software and machine setup

We used Python version 3.6 with frameworks like PyTorch, Pandas, and NumPy. We implemented the experiments such that the local model training at the clients and global model testing at the server happen on a Nvidia Tesla M60 GPU with 8GB of RAM.

4.3. Baselines

We use the below baselines based on TM1 on their relevance and applicability in evasion attacks within FL.

- **FAT [66]:** In adversarial training, the defender generates attack data and augments it with the original normal data to train the model. Federated adversarial training (FAT) combines FL and adversarial training to mitigate the threat of evasion attacks during inference. We add one adversarial sample per training batch of data for every client in our experimentation, which gave the best results.
- **Randomized Smoothing (RS) [9]:** Randomized smoothing is a method for constructing a new, "smoothed" classifier from an arbitrary base classifier. This method tries to turn any classifier that is certifiably robust to adversarial samples under L_2 norm by adding some Gaussian noise to the training and test data. We consider the suggested configuration $\varsigma = 0.1$ in our experimentation.

We use the below baselines based in TM2 against MIA attacks, as stated in [27].

- **Laplacian Noise [54]:** Adds noise to the intermediate activation. The noise follows a Laplacian distribution parameterized by scale b (location parameter μ is kept at 0).
- **Dropout [18]:** Utilizes a mask where each element takes the value 0 with probability p and 1 otherwise. Multiplies the mask element-wise with the intermediate activation.
- **Topk-Prune [61]:** Preserves the top k percent elements in the intermediate activation. Multiplies the mask element-wise with the intermediate activation.
- **Adversarial Noise [59]:** Crafts adversarial noise using Fast Gradient Sign Method (FGSM) [12]. Applies FGSM on a surrogate inversion model (specifically, $L3$ inversion model). Adds the crafted noise to the intermediate activation, scaling the gradient's sign using the symbol ϵ.
- **DistCorr [57]:** Utilizes information correlation techniques such as mutual information and distance correlation. Applies regularization to the training process to enhance the model's resistance against adversarial attacks.
- **Bottleneck Layers [10]**: This method enhances the effectiveness of deep neural networks by minimizing feature sizes. It introduces a partitioning scheme incorporating a bottleneck unit, significantly diminishing the communication costs associated with transferring features between mobile devices and the cloud.

Performance on CIFAR100 dataset. FCD performance on CIFAR100 is already provided in the main paper (Table 6 & Figure 4) against **model inversion attacks**. Now, we also conducted an experiment of FCD under **an evasion utility attack** on CIFAR100. The results in Table 5 demonstrate FCD's consistent superiority across different attack percentages (A_p).

Table 5. Comparison of utility impact ($U \downarrow$) results with $n = 100$, $m = 40$ for CIFAR100 dataset. **ND** denotes no defense.

$A_p \rightarrow$	30%	50%	100%
ND	26.97±0.67	41.06±0.20	48.11±1.96
FAT	19.21±0.17	25.09±0.45	31.90±1.39
RS	22.43±0.26	33.51±1.59	35.18±1.72
FCD (ours)	**13.55±0.32**	**18.80±1.31**	**23.16±1.54**

Performance of local teacher models. Table 6 illustrates the average test accuracy of local client teacher models trained on original data shards distributed among clients for homogeneous and heterogeneous FL settings across four datasets. As anticipated, the teacher models perform better when trained on the original dataspace. However, a slight decrease in accuracy is observed under heterogeneous FL settings, attributed to the non-IID data distribution among clients. These proficient teacher models play a crucial role in knowledge transfer, guiding the training of local student models on FCD-encrypted data space in each round. This not only enhances the overall accuracy but also fortifies the models against challenges posed by TM1 and TM2.

Table 6. Average clients' local **teacher model accuracy** on global test data for homogeneous **(Hom)** and heterogeneous **(Het)** FL settings on five datasets **under no attack**. All values are percentages.

Dataset	Total clients, n	Hom	Het
GTSRB [51]	3	99.82 ±0.42	98.36 ±0.21
	5	98.82 ±0.61	98.14 ±0.56
	10	98.43 ±0.89	98.01 ±0.11
	15	98.39 ±0.79	97.48 ±0.16
	25	97.20 ±0.38	96.98 ±0.20
KBTS [32]	3	98.42 ±0.34	98.36 ±0.87
	5	98.27 ±0.77	98.14 ±0.92
	10	97.85 ±0.80	97.44 ±0.94
	15	97.87 ±0.33	97.18 ±0.67
	25	97.06 ±0.69	96.21 ±0.30
CIFAR10 [23]	100	94.37 ±0.58	93.36 ±0.39
EMNIST [8]	10000	93.50 ±0.36	91.42 ±0.41

A_g of FCD integrated FL under TM2. Table 7 presents the performance of the FCD-integrated FL system under TM2, where the honest-but-curious (HbC) server executes a

Table 7. Comparison of A_g (↑) of FCD with other methods **under TM2** for the CIFAR100 dataset. **ND** denotes no defense.

Defense method	A_g
ND	68.5
Laplacian [54]	58.4
Dropout [18]	57.8
TopkPrune [61]	50.4
AdvNoise [59]	62.0
DistCorr [57]	62.1
Bottleneck Layers [10]	58.0
ResSFL [27]	**67.5**
FCD (ours)	62.3

model inversion attack. FCD operates in an encrypted data space, demonstrating higher Mean Squared Error (MSE), as discussed in the main paper. The global test accuracy A_g of FCD is compared to other methods. While FCD's performance closely aligns with the accuracy of no defense, ResSFL outperforms in terms of utility. However, FCD excels in balancing this tradeoff between utility and privacy, maintaining competitive accuracy while providing robust resilience to MSE attacks. The inherent tradeoff between utility and privacy is effectively managed by FCD, ensuring elevated privacy levels alongside reasonable utility performance, as depicted in Table 7.

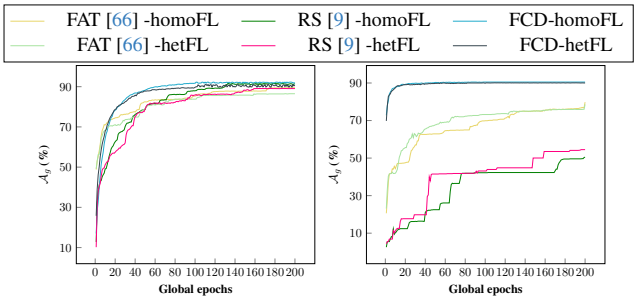
Distillation loss guided by KL Divergence. Table 8 illustrates the performance of our FCD defense under a no attack scenario, both with and without the inclusion of distillation loss ($\mathcal{L}_{KLD}$) guided by Kullback-Leibler (KL) divergence, across four datasets. Remarkably, FCD with $\mathcal{L}_{KLD}$ in the total loss exhibits higher accuracy compared to the scenario without $\mathcal{L}_{KLD}$, showcasing an gain of $3-4\%$. This improvement is attributed to the capability of $\mathcal{L}_{KLD}$ to transfer knowledge from a local teacher model trained on normal data to the local student model, training on FCD-encrypted data space. This property facilitates reduced training loss and enhances model convergence.

Table 8. Comparison of FCD integrated FL system with/without distillation loss for homogeneous **(Hom)** and heterogeneous **(Het)** FL settings on four datasets, in terms of best global test accuracy ($\mathcal{A}_g\%$) ↑ **under no attack**. All values are percentages. Result indicate best results.

	GTSRB [51]		**KBTS [32]**		**CIFAR10 [23]**		**EMNIST [8]**	
Method	**Hom**	**Het**	**Hom**	**Het**	**Hom**	**Het**	**Hom**	**Het**
FCD without $\mathcal{L}_{KLD}$	94.88$_{\pm 0.20}$	93.54$_{\pm 0.24}$	95.74$_{\pm 0.89}$	95.18$_{\pm 0.32}$	73.36$_{\pm 1.51}$	74.62$_{\pm 1.62}$	83.35$_{\pm 0.68}$	80.37$_{\pm 1.69}$
FCD with $\mathcal{L}_{KLD}$	**97.72**$_{\pm 0.43}$	**96.68**$_{\pm 0.18}$	**97.43**$_{\pm 0.12}$	**96.80**$_{\pm 0.46}$	**77.28**$_{\pm 1.77}$	**76.16**$_{\pm 0.19}$	**85.56**$_{\pm 2.12}$	**83.16**$_{\pm 2.08}$

4.4. Time-series Analysis

Figure 3 shows the comparison of test accuracy under attack for each communication round under homogeneous and heterogeneous FL settings on the GTSRB and KBTS datasets, respectively. Here, we show results on a case with $n = 10$ and $A_p = 100$ for brevity. On the GTSRB dataset, it takes about 55 rounds to reach stability as the server is busy aggregating new gradient information from the clients. However, convergence in the homogeneous setting is slightly faster than in the heterogeneous setting because of the regular data samples in each local data shard. On the other hand, FAT and RS take larger rounds for convergence and perform less compared to our FCD. This is because FAT and RS rely on adding perturbed data in terms of adversarial samples and Gaussian noise and hence require many epochs for model convergence. Further, the results on the KBTS dataset are much more diverse. Limited data availability is another reason for the poor convergence performance of FAT and RS. However, our FCD shows stable performance under less data availability and converges quickly compared to other methods.



(a) GTSRB (b) KBTS

Figure 3. Time series comparison of test accuracies under attack for various defenses and FL settings. We show a case with $n = 10$ and $A_p = 100\%$ for brevity.

4.5. Impact of Weight Factor α_k on A_g

We conducted an experiment to examine the impact of the weight factor, α_k, on the global test accuracy A_g of the FCD-integrated FL system. This weight factor allows us to assign significance to the distillation loss guided by KL divergence, which is added to the cross-entropy loss to form the total loss. A larger weight factor corresponds to a higher emphasis on distillation loss. We selected weight factor values within the $\alpha_k \in [0, 2]$ range, specifically 0.1, 0.2, 0.5, 1.5, and 2, and present the global test accuracy without attack in Figure 4. The experiment involved configurations with $n = 25, 25, 100, 10000$ and $m = 25, 25, 70, 500$ for the GTSRB, KBTS, CIFAR10, and EMNIST datasets, respectively, across homogeneous and heterogeneous FL settings. Our observations reveal that, across all cases, setting $\{\alpha_k\}_{k=1}^{\mathcal{N}_k} = 0.5$ yielded the highest A_g. This outcome can be attributed to the balanced weighting of distillation loss and cross-entropy loss, facilitating learning from local teacher models trained on the original data space and

transferring knowledge to local student models trained on FCD-encrypted data space.

5. Discussion

Case about model obfuscation: Correct gradients of the loss function for the input are required for optimization-based attack strategies. One of the main reasons that adaptive attacks [6] are successful is that gradients can be approximated as defensively transformed input is similar to the original input ($T(\mathcal{X}) \approx \mathcal{X}$, where $T(.)$ is some defensive transformation). Hence, evaluating adversarial defense must prevent being lulled into a false sense of security. In contrast, in the proposed FCD method, the input is transformed systematically with a secret key in a simple manner. The resulting input differs from the initial input (i.e., $\mathcal{E}(\mathcal{X}) \not\approx \mathcal{X}$). Key $\mathcal{K}$ directly controls the gradients, unlike conventional obfuscation methods [2]. The pattern created by the FCD transformation with a secret-key $\mathcal{K}$ preserves the gradients of the loss function for the given parameters. Hence, FCD can be viewed as a hard-obfuscating defense.

References

[1] Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8):2864, 2020. 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. ICML, 2018. 10

[3] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017. 1

[4] David Byrd and Antigoni Polychroniadou. Differentially private secure multi-party computation for federated learning in financial applications. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–9, 2020. 1

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020. 1

[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 10

[7] Yu Chen, Fang Luo, Tong Li, Tao Xiang, Zheli Liu, and Jin Li. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Information Sciences*, 522:69–79, 2020. 1

[8] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 7, 8, 9

[9] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019. 1, 8, 9

[10] Amir Erfan Eshratifar, Amirhossein Esmaili, and Massoud Pedram. Bottlenet: A deep learning architecture for intelligent mobile cloud computing services. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6. IEEE, 2019. 8, 9

[11] Ronald Aylmer Fisher and Frank Yates. *Statistical tables for biological, agricultural, and medical research*. Hafner Publishing Company, 1953. 5

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 8

[13] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018. 1

[14] Hanxi Guo, Hao Wang, Tao Song, Yang Hua, Zhangcheng Lv, Xiulang Jin, Zhengui Xue, Ruhui Ma, and Haibing Guan. Siren: Byzantine-robust federated learning via proactive alarming. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 47–60, 2021. 1

[15] Ehsan Hallaji, Roozbeh Razavi-Far, Mehrdad Saif, and Enrique Herrera-Viedma. Label noise analysis meets adversarial training: A defense against label poisoning in federated learning. *Knowledge-Based Systems*, page 110384, 2023. 1

[16] Bedros Hanounik and Xiaobo Hu. Linear-time matrix transpose algorithms using vector register file with diagonal registers. In *Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*, pages 8–pp. IEEE, 2001. 5

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 7

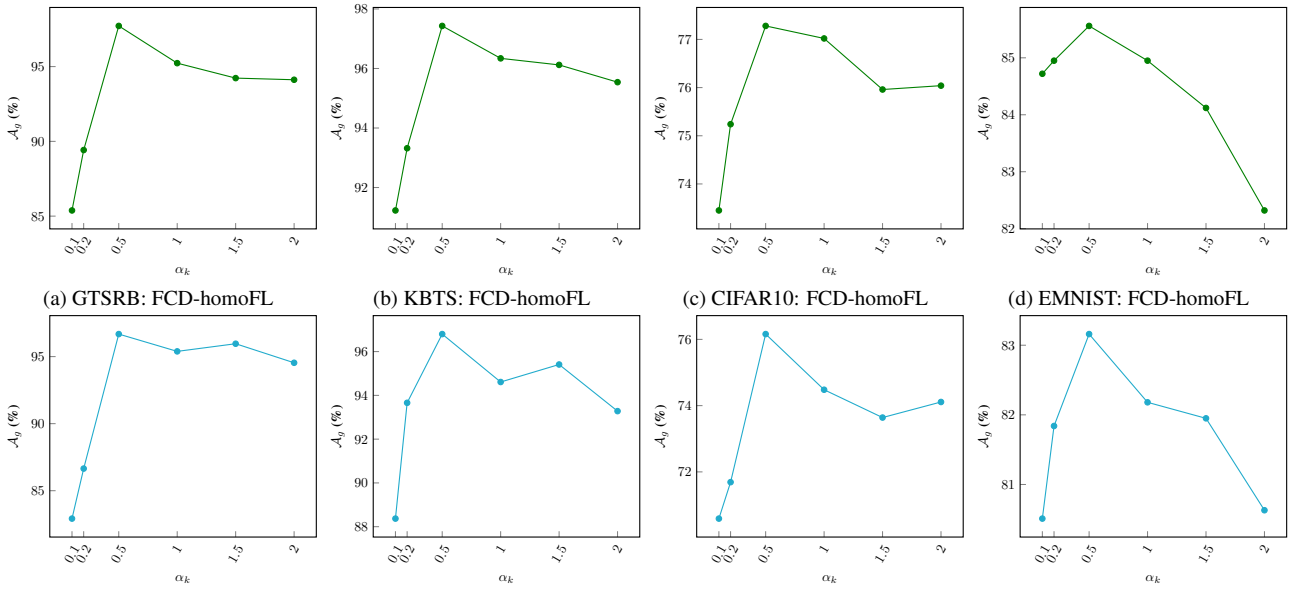
[18] Zecheng He, Tianwei Zhang, and Ruby B Lee. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet of Things Journal*, 8(12):9706–9716, 2020. 3, 8, 9

[19] Neveen Mohammad Hijazi, Moayad Aloqaily, Mohsen Guizani, Bassem Ouni, and Fakhri Karray. Secure federated learning with fully homomorphic encryption for iot communications. *IEEE Internet of Things Journal*, 2023. 5

[20] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 7(10):9530–9539, 2020. 1

[21] Najeeb Moharram Jebreel and Josep Domingo-Ferrer. Fl-defender: Combating targeted attacks in federated learning. *Knowledge-Based Systems*, 260:110178, 2023. 1

[22] Zoe L Jiang, Hui Guo, Yijian Pan, Yang Liu, Xuan Wang, and Jun Zhang. Secure neural network in federated learning with model aggregation under multiple keys. In *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 47–52. IEEE, 2021. 5



(a) GTSRB: FCD-homoFL (b) KBTS: FCD-homoFL (c) CIFAR10: FCD-homoFL (d) EMNIST: FCD-homoFL

(e) GTSRB: FCD-hetFL (f) KBTS: FCD-hetFL (g) CIFAR10: FCD-hetFL (h) EMNIST: FCD-hetFL

Figure 4. Effect of the total loss weight factor α_k on the performance of FCD integrated FL system across diverse datasets is illustrated, considering both homogeneous and heterogeneous FL settings **in the absence of any attack**. The settings involve $n = 25, 25, 100, 10000$ and $m = 25, 25, 70, 500$ for the GTSRB, KBTS, CIFAR10, and EMNIST datasets, respectively.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7, 8, 9

[24] K Naveen Kumar, C Vishnu, Reshmi Mitra, and C Krishna Mohan. Black-box adversarial attacks in autonomous vehicle technology. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7. IEEE, 2020. 3, 4

[25] K Naveen Kumar, C Krishna Mohan, and Linga Reddy Cenkeramaddi. The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[27] Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10194–10202, 2022. 7, 8, 9

[28] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. 6

[29] Xiaoyuan Liu, Hongwei Li, Guowen Xu, Zongqi Chen, Xiaoming Huang, and Rongxing Lu. Privacy-enhanced federated learning against poisoning adversaries. *IEEE Transactions on Information Forensics and Security*, 16:4574–4588, 2021. 1

[30] M lTSU RUNAKAI and TOSH IM ASA TADA. The reverse triangle inequality in normed spacest. *NEW ZEALAND JOURNAL OF MATHEMATICS*, 25:181–193, 1996. 7

[31] Zhuoran Ma, Jianfeng Ma, Yinbin Miao, Yingjiu Li, and Robert H Deng. Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning. *IEEE Transactions on Information Forensics and Security*, 17:1639–1654, 2022. 1

[32] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition — how far are we from the solution? In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013. 7, 8, 9

[33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 6

[34] Thomas Minka. Estimating a dirichlet distribution, 2000. 2

[35] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pages 94–108, 2021. 1

[36] Arup Mondal, Yash More, Ruthu Hulikal Rooparaghunath, and Debayan Gupta. Poster: Flatee: Federated learning across trusted execution environments. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 707–709. IEEE, 2021. 1

[37] Maria Moszyńska and Wolf-Dieter Richter. Reverse triangle inequality. antinorms and semi-antinorms. *Studia Scien-*

tiarum Mathematicarum Hungarica, 49(1):120–138, 2012. 7

[38] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. Smpai: Secure multi-party computation for federated learning. In *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019. 1

[39] Muslim Muslim, Yulita Salim, Erick Irawadi Alwi, Huzain Azis, et al. Modified transposition cipher algorithm for images encryption. In *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pages 1–4. IEEE, 2018. 3

[40] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pages 7587–7624. PMLR, 2022. 1

[41] Jaehyoung Park, Nam Yul Yu, and Hyuk Lim. Privacy-preserving federated learning using homomorphic encryption with different encryption keys. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1869–1871. IEEE, 2022. 3

[42] Michael R Portnoff. An efficient parallel-processing method for transposing large matrices in place. *IEEE Transactions on Image Processing*, 8(9):1265–1275, 1999. 5

[43] Vike Maylana Putrie, Christy Atika Sari, Eko Hari Rachmawanto, et al. Super encryption using transposition-hill cipher for digital color image. In *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 152–157. IEEE, 2018. 3

[44] Abhijeet A Ravankar and Stanislav G Sedukhin. An o (n) time-complexity matrix transpose on torus array processor. In *2011 Second International Conference on Networking and Computing*, pages 242–247. IEEE, 2011. 5

[45] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. *arXiv preprint arXiv:2201.00763*, 2022. 1

[46] Devansh Shah, Parijat Dube, Supriyo Chakraborty, and Ashish Verma. Adversarial training in communication constrained federated learning. *arXiv preprint arXiv:2103.01319*, 2021. 1

[47] Muhammad Shayan, Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Biscotti: A blockchain system for private and secure federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1513–1525, 2020. 1

[48] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, 2022. 2, 3

[49] Young Ah Shin, Geontae Noh, Ik Rae Jeong, and Ji Young Chun. Securing a local training dataset size in federated learning. *IEEE Access*, 10:104135–104143, 2022. 3

[50] Massoud Sokouti, Babak Sokouti, and Saeid Pashazadeh. An approach in improving transposition cipher system. *Indian Journal of Science and Technology*, 2(8):9–15, 2009. 3

[51] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pages 1453–1460, 2011. 7, 8, 9

[52] Nurbek Tastan and Karthik Nandakumar. Capride learning: Confidential and private decentralized learning based on encryption-friendly distillation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8084–8092, 2023. 3

[53] Jeyarajan Thiyagalingam, Olav Beckmann, and Paul HJ Kelly. An exhaustive evaluation of row-major, column-major and morton layouts for large two-dimensional arrays. In *Performance Engineering: 19th Annual UK Performance Engineering Workshop*, pages 340–351. University of Warwick Coventry, UK, 2003. 4

[54] Tom Titcombe, Adam J Hall, Pavlos Papadopoulos, and Daniele Romanini. Practical defences against model inversion attacks for split neural networks. *arXiv preprint arXiv:2104.05743*, 2021. 8, 9

[55] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2587–2596. IEEE, 2019. 1

[56] Dmitrii Usynin, Alexander Ziller, Marcus Makowski, Rickmer Braren, Daniel Rueckert, Ben Glocker, Georgios Kaissis, and Jonathan Passerat-Palmbach. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence*, 3(9): 749–758, 2021. 3

[57] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 933–942. IEEE, 2020. 8, 9

[58] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020. 1

[59] Jing Wen, Siu-Ming Yiu, and Lucas CK Hui. Defending against model inversion attack by adversarial examples. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 551–556. IEEE, 2021. 8, 9

[60] Zuobin Xiong, Zhipeng Cai, Daniel Takabi, and Wei Li. Privacy threat and defense for federated learning with non-iid data in aiot. *IEEE Transactions on Industrial Informatics*, 18(2):1310–1321, 2021. 1

[61] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 225–240, 2019. 8, 9

[62] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 1

[63] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020. 1

[64] Chi Zhang, Sotthiwat Ekanut, Liangli Zhen, and Zengxiang Li. Augmented multi-party computation against gradient leakage in federated learning. *IEEE Transactions on Big Data*, 2022. 1

[65] Li Zhang, Jianbo Xu, Pandi Vijayakumar, Pradip Kumar Sharma, and Uttam Ghosh. Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system. *IEEE Transactions on Network Science and Engineering*, 2022. 3, 5

[66] Giulio Zizzo, Ambrish Rawat, Mathieu Sinn, and Beat Buesser. Fat: Federated adversarial training. *arXiv preprint arXiv:2012.01791*, 2020. 1, 8, 9

[67] Tianyuan Zou, Yang Liu, Yan Kang, Wenhan Liu, Yuanqin He, Zhihao Yi, Qiang Yang, and Ya-Qin Zhang. Defending batch-level label inference and replacement attacks in vertical federated learning. *IEEE Transactions on Big Data*, 2022. 1