electronics

Article

Unmanned Aerial Vehicle-Assisted Federated Learning Method Based on a Trusted Execution Environment

Jia Liao [1,2], Baihui Jiang [1,2], Peng Zhao [1], Lei Ning [1,*] and Liming Chen [3]

[1] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
[2] College of Applied Technology, Shenzhen University, Shenzhen 518060, China
[3] Electric Power Research Institute, China Southern Power Grid CSG, Guangzhou 510663, China
* Correspondence: ninglei@sztu.edu.cn

Abstract: In the face of increasing concerns around privacy and security in the use of unmanned aerial vehicles (UAVs) for mobile edge computing (MEC), this study proposes a novel approach to secure UAV-assisted federated learning. This research integrates a trusted execution environment (TEE) into UAV-assisted federated learning and proposes a robust aggregation algorithm based on cosine distance, denoted as CosAvg. This study further designs and evaluates a TEE-based federated learning model, comparing its resource overhead with other secure aggregation frameworks, like homomorphic encryption (HE) and differential privacy (DP). Experimental results indicate a significant reduction in resource overhead for TEE against DP and HE. Moreover, the proposed CosAvg algorithm demonstrated superior robustness against adversarial scenarios, maintaining high accuracy in the presence of malicious clients. The integration of TEE and the CosAvg algorithm provides a secure and robust solution for UAV-assisted federated learning, effectively defending both gradient inversion attacks and byzantine attacks.

Keywords: unmanned aerial vehicle; federated learning; trusted execution environment; assisted learning; mobile edge computing

check for updates

Citation: Liao, J.; Jiang, B.; Zhao, P.; Ning, L.; Chen, L. Unmanned Aerial Vehicle-Assisted Federated Learning Method Based on a Trusted Execution Environment. *Electronics* **2023**, *12*, 3938. https://doi.org/10.3390/electronics12183938

Academic Editor: Shiho Kim

Received: 31 August 2023
Revised: 8 September 2023
Accepted: 14 September 2023
Published: 18 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have demonstrated considerable potential in the field of mobile edge computing, capitalizing on their remarkable flexibility and mobility [1,2]. However, UAVs often perform tasks in open environments, which may involve the collection and processing of sensitive data [3]. Consequently, concerns pertaining to privacy and security come to the forefront. As a solution, the concept of UAV-assisted federated learning [4,5] has emerged, offering a novel approach to address this challenge.

In a federated learning model with UAVs as the aggregation servers [6], each edge device acts as a participant in the federated learning process. They locally train machine learning models and upload gradients to the UAV. Subsequently, the UAV performs aggregation calculations on these gradient parameters and distributes the aggregated gradient values back to the edge devices for further training. Through multiple rounds of iteration, the UAV can obtain a new global model without knowledge of the training dataset [7].

Compared to conventional cloud servers, the advantages of UAV-aggregated servers lie in their ability to optimize participant communication latency through dynamic spatial adjustments [8]. Moreover, UAVs are not reliant on fixed communication network infrastructure and can deploy and execute federated learning tasks in intricate geographical environments [9].

However, existing research indicates security issues within federated learning. (1) Hitaj et al. [10] highlighted a security vulnerability known as gradient inversion attacks. This vulnerability arises from the fact that an aggregation server can exploit model parameters explicitly uploaded by a participant to reconstruct a portion of their training dataset. This

clearly contradicts the fundamental goal of safeguarding participant privacy in federated learning. (2) Roszel et al. [11] revealed the presence of a byzantine problem within federated learning. Malicious client-side devices can transmit fabricated gradient information to undermine the accuracy of the final aggregated model. These security challenges are further exacerbated in the context of UAV-assisted federated learning. The susceptibility of UAVs to physical compromise due to their exposure to open physical environments enhances the vulnerability of the system to attacks. This issue emerges when a malicious client-side devices transmit fake gradient information, thereby compromising the integrity of the final aggregated model. Furthermore, UAVs communicate through open wireless networks, significantly reducing the cost and difficulty for attackers to launch Byzantine attacks.

Therefore, in response to the aforementioned issues, researchers have proposed several approaches to address these problems. (1) Regarding the first issue, three privacy-preserving techniques have been introduced to enhance the security of federated learning: secure multi-party computation (MPC) [12], homomorphic encryption (HE) [13], and differential privacy (DP) [14,15]. Among them, HE and MPC can effectively prevent gradient inversion attacks, but they generate significant computational overhead and expensive communication costs, making them unsuitable for resource-constrained UAVs. DP techniques are efficient; however, they can reduce the accuracy and performance of the federated learning model's aggregation. Overall, existing solutions face a trade-off between model accuracy and efficiency, thereby hindering their adoption. (2) To address the second issue, new federated learning model aggregation algorithms have been proposed to enhance robustness against malicious attackers. Algorithms like Krum [16] calculate the Euclidean distance between each model to exclude malicious clients; the RFA [17] algorithm replaces weighted arithmetic averages with approximated geometric medians for aggregation and demonstrates robustness under high pollution scenarios. However, The robustness of these algorithms is limited and could potentially come with increased computational overhead. Therefore, these algorithms are not suitable for UAV-assisted federated learning tasks.

Recent research studies [18,19] suggest that trusted execution environments (TEEs) are promising and efficient at mitigating privacy leaks in federated learning. TEE provides an isolated execution environment separated from the host environment to protect sensitive data and critical code from unauthorized access and attacks. Even if the operating system or applications are compromised by malicious software, attackers cannot bypass solidified hardware logic and hardware-level tampering detection. Thus, TEE remains independent and trustworthy. By performing encryption and aggregation operations within the TEE, the confidentiality and integrity of parameters within UAVs can be ensured. Additionally, TEE encryption leverages hardware support, minimizing processor and memory resource usage, resulting in higher performance during encryption operations. Compared to federated learning security solutions based solely on cryptographic algorithms mentioned above, combining TEE with UAV offers improved security and performance.

Therefore, we propose a UAV-assisted federated learning model based on TEE. Considering the limited computing performance and energy of UAVs. Model parameters are encrypted before being uploaded to UAVs' secure zone to prevent parameter leakage. The UAV then performs decryption and secure aggregation within the secure zone. To further enhance the efficiency of this model, a robust aggregation algorithm based on cosine distance is introduced, which improves the model's resilience against byzantine attacks [20] while maintaining aggregation efficiency. By utilizing TEE hardware security measures to protect the security of federated learning and combining them with efficient robust aggregation algorithms, this model can enhance the security and performance of UAV-assisted federated learning. The main contributions of our study are as follows:

- **We introduce TEE into UAV-assisted federated learning**. We employ TEE to safeguard the security of model parameter aggregation in UAV-assisted federated learning. The UAV assumes the role of an aggregation server, executing decryption and aggregation within its trusted zone. This ensures the parameters' confidentiality, integrity, and effectiveness toward gradient inversion attacks.

- **We propose a robust aggregation algorithm based on cosine distance**. To combat the challenges posed by byzantine attacks in UAV federated learning, we present an efficient, robust aggregation algorithm called CosAvg. This algorithm excludes malicious clients, enhances model stability against byzantine attacks, and maintains aggregation efficiency.
- **We design and evaluate a federated learning model based on TEE**. We construct a federated learning model for aggregation within the TEE of the aggregation server. We also compare the resource overhead of TEE against other secure aggregation frameworks, including HE and DP. We finally conduct performance comparisons between our proposed aggregation algorithm and other robust aggregation algorithms. Experimental results indicate that the proposed aggregation algorithm exhibits higher robustness.

The rest of our study is structured as follows: Section 2 introduces the background knowledge of TEE, then discusses two security issues faced by federated learning: gradient inversion attacks and byzantine attacks. Section 3 provides a comprehensive exposition of TEE framework designed to counter gradient inversion attacks, alongside the presentation of the CosAvg aggregation algorithm tailored to mitigate byzantine attacks. Section 4 presents the effectiveness of both the TEE framework and the CosAvg algorithm through two experiments. Section 5 offers a comprehensive summary of our study.

2. Related Work

2.1. TEE and TrustZone

A trusted execution environment (TEE) is a new hardware security feature that is isolated from a normal OS (i.e., rich execution environment (REE)) [21]. The design objective of TEE is to provide a highly trusted execution environment that can protect sensitive information and critical computations even when the host operating system has been invaded. TrustZone is the implementation of TEE on the ARM architecture. It divides the processor into two isolated regions, namely the secure world and the normal world.

REE and TEE are mutually isolated. The TEE architecture is illustrated in Figure 1. Sensitive data are typically stored in encrypted form within the REE. It is then transferred to the TEE side through the trusted application (TA) API for decryption and processing within the TEE. Finally, the results are returned to the REE. Due to the inaccessibility of the TEE to other programs, the processing of sensitive data can be ensured to remain confidential and protected from unauthorized access.

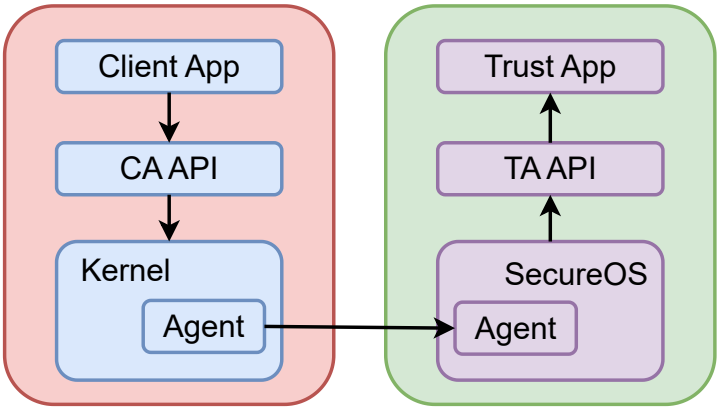


Figure 1. TEE architecture. It explains how TEE safeguards the security of sensitive data.

Alves et al. [22] introduced for the first time the application of ARM TrustZone within embedded systems. Liu et al. [23] innovatively incorporated ARM TrustZone, employing a trusted computational block to safeguard peripheral devices from unauthorized access by malicious applications. However, this research does not delve deeply into the specific task of federated learning. Zhang et al. [24] explored the utilization of multiple UAVs for

federated learning tasks. However, their study did not introduce TEE to address the security concerns of federated learning. Mo et al. [25] proposed a privacy-preserving federated learning (PPFL) framework that leverages mobile devices with TEE environments to curtail privacy breaches in federated learning. However, this framework is not tailored specifically for UAVs.

2.2. Gradient Inversion Attacks

Within the entire life-cycle of federated learning, a variety of attacks and threats emerge. Among these, attacks during the model training phase stand out as a primary research focus in ensuring the security of federated learning. Adversaries hold the capacity to compromise either the central server or the clients, thereby eavesdropping on and pilfering transmitted parameters. One notably prominent attack is the gradient inversion attack, which aims to reconstruct or recover sensitive information from the shared gradients.

Zhu et al. [26] introduced the concept of deep leakage from gradient (DLG), wherein the fundamental principle revolves around optimizing synthetic gradients that closely resemble the original gradients. This optimization aims to ensure that the generated synthetic data approximates real training data. In a similar vein, Li et al. [27] proposed the notion of generative gradient leakage (GGL). This approach leverages the latent space of generative adversarial networks (GANs) learned from publicly available image datasets as a prior, thereby compensating for the information loss during the gradient degradation process.

Furthermore, Wang et al. [28] formulated the approach of gradient-based adaptive privacy attack (SAPAG), wherein the distance metric is established based on gradient differences using a Gaussian kernel. Zhu et al. [29] devised R-GAP, presenting a recursive procedure for data recovery from gradients. In a different context, Jin et al. [30] introduced a catastrophic data leakage attack within the framework of vertical federated learning (CAFE). This method aims to execute large-scale data leakage attacks under the vertical federated learning setting while concurrently enhancing the quality of data recovery processes.

A straightforward strategy for countering adversarial inversion attacks involves encrypting the gradients. Hardware-based methods encompass the TEE mentioned earlier, while software-based approaches include HE. For instance, Phong et al. [13] proposed the utilization of HE to encrypt gradients before transmission. Zhang et al. [31] introduced an efficient HE solution called BatchCrypt for cross-private-key federated learning, effectively mitigating the communication overhead induced by encryption.

Beyond encryption, noise-based solutions can be employed, entailing the introduction of noise to gradients to perturb them, a concept termed differential privacy (DP). For instance, Wei et al. [32] presented a versatile framework that integrates FL with DP, ensuring distinct protection levels by tuning diverse levels of noise. McMahan et al. [33] introduced a novel algorithm, termed DP-FedAvg, designed for user-level DP training of large-scale neural networks within a federated setting. However, both HE and DP inherently impact model performance, thus striking a balance between security and utility remains a challenging endeavor [34].

2.3. Byzantine Attacks

Byzantine attacks are poisoning attacks in which malicious participants in a multi-node system intentionally provides incorrect information, with the aim of degrading the performance of the model. Shi et al. [35] proposed a method called the weight attack, which aims to hide the attacker's dataset size and change the model weights during model aggregation. Fang et al. [36] introduced the concept of localized model poisoning attack, which involves manipulating the local models uploaded from compromised devices to the central server during the training process.

The basic aggregation algorithm FedAvg [37] is vulnerable to byzantine attacks, and as a result, researchers have proposed several secure and robust aggregation algorithms. Blanchard et al. [16] proposed the Krum aggregation rule, which selects the model most similar to other models from a set of local models as the global model. Krishna et al. [17]

introduced the RFA algorithm, which uses the weighted median as the parameter for the global model. Mhamdi et al. [38] proposed the Bulyan algorithm, which combines the Krum aggregation algorithm with the trimmed mean aggregation algorithm. Bulyan applies the Krum aggregation algorithm iteratively to select the local models and then aggregates them using the trimmed mean aggregation algorithm to obtain the global model. This method helps mitigate the influence of certain abnormal model parameters in the Krum aggregation algorithm. Fung et al. [39] proposed the FoolsGold algorithm, which employs the concept of cosine similarity to identify the contributions of malicious clients. When unusually high similarity is detected, the aggregation server adjusts the contributions of these clients by utilizing a lower learning rate, resulting in updates along the direction of decreasing reverse gradients.

Therefore, our research is dedicated to protecting the privacy and enhancing the robustness of federated learning with minimal computational cost, aiming to improve the efficiency and security of UAV-assisted federated learning.

3. Approach

3.1. Architecture for UAV-Assisted Federated Learning Based on TEE

In the context of federated learning tasks conducted by UAVs, clients may encompass a variety of edge devices, such as smartphones and autonomous vehicles. These diverse clients employ varying data preprocessing methods, and engage in localized training of machine learning models [40–42].

Subsequently, they transmit updated model parameters to the UAV for aggregation after each round. Subsequently, they receive the model provided by the server and continue the training process. This iterative process continues until a final federated learning model is trained and established. Based on this, our system architecture is designed as Figure 2.

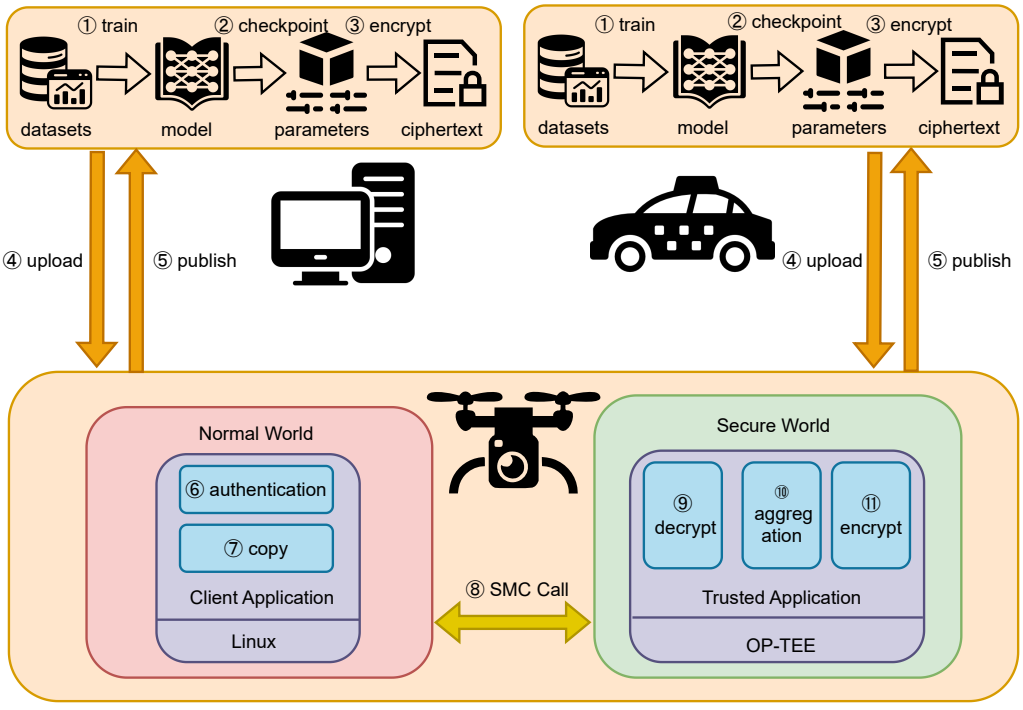


Figure 2. System architecture. This figure illustrates the collaborative process of federated learning between UAV aggregation servers and mobile edge devices, with a particular emphasis on the UAV aggregation process within the TrustZone.

Firstly, the UAV securely stores digital certificates within TEE and sends them to all clients for client-to-UAV authentication. Upon successful verification, clients send their embedded digital certificates to the UAV for UAV-to-client authentication. This establishes

mutual authentication. Subsequently, the UAV and mobile edge devices engage in an elliptic curve Diffie–Hellman (ECDH) algorithm to negotiate a session key, which is then used for encrypting model parameters. Within the secure world, the UAV maintains a mapping of each client's identity to their respective session key.

Next, as a federated learning server, the UAV initializes a model that is broadcast to all clients. To safeguard against potential model parameter privacy leaks during transmission or within the UAV's normal world, the shared model is initialized within the secure world and subsequently encrypted. Clients receive the model, decrypt it, and then locally train it with their respective datasets. After several rounds of training, the model is AES-GCM-encrypted using the pre-negotiated session key and sent back to the UAV.

Upon receiving the encrypted models from clients, the UAV temporarily stores them in the normal world and waits for successful uploads from all clients. Due to the limited memory within the TEE, only a specified number of layers (N layers) are transmitted to the secure world at a time. Within the secure world, the model is decrypted using the session keys of each client, yielding the original model parameters. Subsequently, the UAV performs aggregation on the plaintext models within the secure world, encrypts the aggregated model using the same keys, and moves it back to the normal world. Once aggregation is completed for all layers, the model is concatenated in the normal world, and the encrypted aggregated model is sent to all clients for further training. This concludes one round of federated learning within the TEE. By repeating this process multiple times, we eventually obtain a global federated learning model that leverages multi-party collaboration.

The pseudo-code for the above algorithm is shown in Algorithm 1.

Algorithm 1 Aggregating process in UAV as the federated learning server.

Input: Model Layers: $\{L_1, L_2, \ldots, L_m\}$;
Number of Model Layers: M;
Clients in FL: $\{C_1, C_2, \ldots, C_n\}$;
Number of clients: N;
Secret keys: $\{SK_1, SK_2, \ldots, SK_n\}$;
Communication rounds: R.
Output: Encrypted Model

for each $r \in [1, R]$ **do**
  receive encrypted model parameter P from clients C_j in round r
  for each $i \in [1, M]$ **do**
    retrieve parameters P_i of the current layer
    V_i = flatten(P_i)
    loading V_i into the secure world
    Secure world:
    for each $j \in [1, N]$ **do**
      W_{ij} = Decode(V_{ij}, SK_j)
      W_i = aggregation(W_{i1}, W_{i2}, …, W_{in})
      H_{ij} = Encode(W_i, SK_j)
    end for
    Move H_i into the normal world
  end for
  send H_{ij} to client C_j
end for

Throughout the entire federated learning process, model parameters are encrypted during transmission and within the non-secure normal world. Additionally, aggregation operations are conducted within the hardware-secured TEE environment, effectively safeguarding model privacy and security. Moreover, since decrypted model parameters exist in plaintext within the TEE, they can be replaced with robust algorithms that are resistant to Byzantine attacks.

3.2. Cosine Distance-Based Aggregation Algorithm

On the one hand, the Euclidean distance is extensively employed for detecting malicious clients, exemplified by algorithms such as Krum and multi-Krum. However, the Euclidean distance is prone to suffering from the "curse of dimensionality", which refers to the challenges and limitations that arise when working with high-dimensional data. The curse of dimensionality would increase computational complexity and decrease the accuracy of clustering. On the contrary, the cosine distance is not impervious to the curse of dimensionality in sparse and discrete high-dimensional space. This implies that in tasks involving clustering based on high-dimensional model parameters, cosine distance may perform better than Euclidean distance.

On the other hand, the concept of the median, which is used by RFA and trimmed–median algorithms, is generally considered to be a more robust aggregation method compared to the concept of the mean, as it can ignore outlier data points. However, the stability of the median virtually relies on a relatively uniform distribution of outlier data, if we first exclude a majority of exceptional clients based on cosine distance, the mean algorithm will exhibit stronger aggregation performance and stability for different data distributions.

Based on the aforementioned discussion, we contend that the combination of cosine distance and the concept of mean may hold stronger theoretical practicality. Therefore, we propose the CosAvg algorithm. The calculation method for the cosine distance is illustrated in Equation (1).

$$D(\mathbf{x}, \mathbf{y}) = cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}} \tag{1}$$

The core concept of our algorithm is to extract the parameter vectors of each client at the current layer within the secure world. Then it calculates the cosine distances between the parameter vectors of each client and those of other clients. These distances serve as a measure of similarity between the clients. If a particular client has a considerable distance from the other clients, it indicates that its similarity with the majority of the models is relatively low. In such cases, we identify this client as a potentially malicious one. The pseudo-code of the CosAvg algorithm is shown in Algorithm 2.

Algorithm 2 Robust aggregation algorithm.

Input: Clients in FL: $\{C_1, C_2, \ldots, C_n\}$;
Parameter vectors in each client: $\{W_1, W_2, \ldots, W_n\}$;
Number of clients: N;
Output: Aggregated Vector W

for each $i \in [1, N]$ **do**
  for each $j \in [i, N]$ **do**
    if $i \neq j$ **then**
      $D(C_i, C_j) = D(C_j, C_i) = (W_i \cdot W_j) / ||W_i|| * ||W_j||$
    end if
  end for
  $S_i = \sum_{k=1, k \neq i}^{n} D(C_i, C_k)$
end for
sort $\{S_1, S_2, \ldots, S_n\}$ into $\{S'_1, S'_2, \ldots, S'_n\}$
W = $\sum_{k=1}^{n-f} S'_k / (n - f)$
return W

As input, it takes a set of clients participating in federated learning. The parameter vector of each client C_i is denoted as W_i. It performs pairwise comparisons among the clients' parameter vectors and calculates normalized inner products to determine the similarity values. Subsequently, it computes a similarity score S_i for each client by summing up the relevant similarity values, excluding self-similarity. These similarity scores are then

sorted, and the algorithm excludes the maximum f clients and then aggregates the top $n - f$ models to generate the final aggregated vector W, which serves as parameter vectors of the global model.

4. Experiment and Result Analysis

In this section, we design two experiments: (1) We first evaluate the CPU time and memory usage of TEE compared with the federal learning security aggregation framework including DP and HE. The results reveal that TEE introduces only a minor additional overhead. It is, therefore, worthwhile to invest a small amount of computational overhead in exchange for the security of federal learning tasks. TEE exhibits higher performance in contrast to other software-based secure algorithms, which suggests that we can implement the security of federated learning with minimal resource expense. (2) We compare the other two robustness algorithms with CosAvg to highlight its better robustness for defending adversarial Byzantine attacks. The results show that TEE can be employed to ensure the security of the aggregation process, while the CosAvg effectively mitigates the Byzantine attack for model aggregation in the security zone of TEE. The proposed approach in our study can be efficiently applied to federated learning scenarios where UAV serves as aggregation servers, achieving a balanced integration of UAV performance, security, and resource consumption.

4.1. Experimental Setup

Datasets: We select two common and simple image classification datasets, namely the MNIST dataset for handwritten digits and the CIFAR-10 dataset for colored images.

MNIST: The MNIST dataset consists of a collection of handwritten digits, each represented as a grayscale image of size 28×28 pixels. The dataset contains a total of 60,000 training images and 10,000 testing images. Each image is labeled with the corresponding digit it represents, ranging from 0 to 9. The MNIST dataset is often used as a simple starting point for experimenting with image classification algorithms.

CIFAR10: The CIFAR10 dataset consists of 60,000 small colored images in 10 different classes, with 6000 images per class. Each image is a color image with dimensions 32×32 pixels. The dataset is split into 50,000 training images and 10,000 testing images. CIFAR10 presents a more challenging classification task compared to MNIST due to its higher resolution, color images, and a wider variety of objects.

Models: We use a simple image classification task to evaluate the performance of UAV-assisted federated learning. Given the constrained memory and computational resources of the UAV, aggregating models with extensive layers and parameters presents a challenge. As a result, we choose to employ two comparatively uncomplicated models: LeNet for MNIST and VGG9 for CIFAR10. Due to limitations in the capacity of the secure world, we place only the first 3 layers of the model within TrustZone for aggregation.

LeNet: The LeNet boasts a compact network architecture frequently utilized for MNIST dataset training. It encompasses two convolutional layers, two pooling layers, and three fully connected layers. The size of the convolutional kernel is 5×5. The activation function is ReLU.

VGG9: The VGG9 is a variant of the VGG network architecture, possessing fewer layers and parameters compared to VGG-16 and VGG-19. VGG9 is comprised of 6 convolutional layers and 3 fully connected layers, The size of the convolutional kernel is 3×3. The activation function is also ReLU.

Hardware: The testing environment encompasses two principal components: an aggregation server and multiple training clients. They are connected within the local area whose network delay is about 60 ms. We employ a Raspberry Pi 3B to simulate the UAV used as a federated learning aggregation server. Raspberry Pi 3B is equipped with an ARM architecture processor, which offers hardware configurations and performance similar to the majority of popular UAVs. Notably, Raspberry Pi also provides ARM TrustZone support. While it is acknowledged that Raspberry Pi fails to implement certain secure features,

such as secure boot, memory, and peripherals, it is still sufficient to utilize Raspberry for simulation and testing purposes. We subsequently installed OP-TEE OS on Raspberry Pi, utilizing its TrustZone interface to implement multiple secure aggregation algorithms running in the secure world. On a separate x86 workstation, we deployed the training clients, simulating multiple federated learning clients through multi-thread programming. This design allows for easy scalability of the client count, thereby facilitating performance evaluation of the aggregation server under varying workloads. Since our experiments primarily concentrate on the TEE module of the federated learning aggregation server, we did not implement TEE support within the client. Table 1 lists detailed configuration information.

Table 1. Detailed configuration information about the aggregation server and the training clients.

Device	Parameter	Value
Raspberry Pi 3B	CPU	ARM Cortex-A53
	Cores	4
	Memory	1GB LPDDR2
	Flash	32GB microSD
	OS	OP-TEE OS 3.12
Workstation	CPU	Intel Xeon Gold 5220
	Cores	24
	GPU	NVIDIA Quadro RTX 5000
	Memory	64 GB
	OS	Ubuntu 20.04.4 LTS

4.2. Overhead across Various Secure Aggregation Frameworks

In this experiment, We analyze the runtime overhead usage of TEE in comparison to other secure frameworks for federated learning, including HE and DP. In order to further assess the overhead imposed by TEE itself, we additionally incorporate a control group named REE, which mirrors the experimental conditions of TEE except for the absence of any security framework.

We train MNIST with LeNet and CIFAR10 with VGG on client-side devices. The trained models are then uploaded to an aggregation server where different security frameworks are applied for aggregation. Because of the limited secure memory within the TEE, we transmit only a subset of the model's layers for aggregation within the secure world, subsequently exchanging them with the normal world. Besides, DP and HE are also executed in the normal world. We define a full federated learning round as the whole period, which starts from the reception of model parameters from all clients, continues with the aggregation of global models, and ends with the distribution of the model to respective clients. We use two performance metrics to quantify the runtime overhead of various security frameworks: The CPU time refers to the average duration taken for a single round, while memory usage indicates the average memory consumed during a round. For the MNIST with LeNet, we set the number of rounds to 200, and for CIFAR10 with VGG9, we set the number of rounds to 500. The aggregation algorithm used is FedAvg. Figure 3 shows the results of CPU time and memory usage of REE, TEE, DP, and HE.

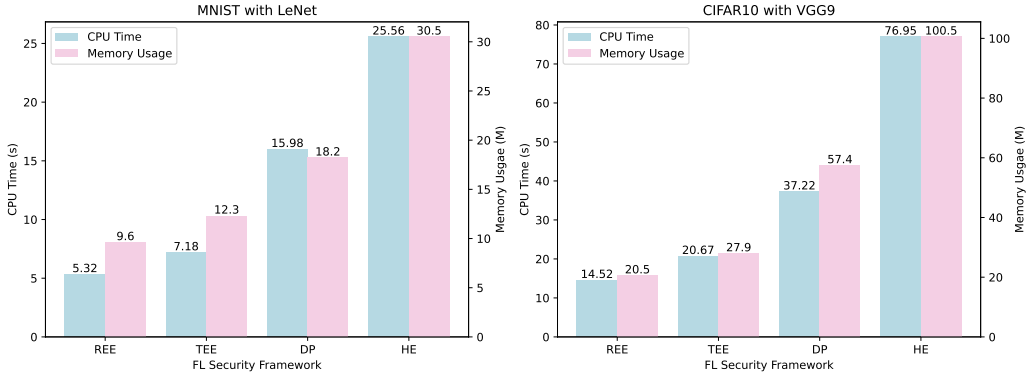


Figure 3. Time and memory overhead with various secure aggregation frameworks.

Compared to DP and HE, it is evident that DP and HE consume higher CPU time and memory usage than TEE. Specifically, DP's resource overhead is roughly twice that of TEE, while HE's overhead reaches three times that of TEE. On the one hand, the noise introduced by DP results in greater computational overhead during model aggregation. It is important to note that the noise may distort the true distribution of data, potentially affecting the accuracy of the global aggregation model. On the other hand, HE involves complex mathematical operations on ciphertext, leading to slower computation during aggregation. Additionally, HE suffers from ciphertext expansion, where encrypted data are significantly larger than plaintext, resulting in higher memory consumption.

Compared to REE, TEE exhibits a 35.0% increase in CPU time and a 28.1% increase in memory usage for MNIST with LeNet. Similarly for CIFAR10 with VGG9, TEE shows a 42.4% increase in CPU time and a 36.1% increase in memory usage. This increase originates from the additional overhead introduced by the TEE security mechanisms, including the mode-switching time between the secure and normal worlds, the model copy overhead from the secure world to the normal world, and encryption/decryption cost for the model. The time overheads for these phases are measured in Figure 4. The transition from LeNet to VGG9 results in additional increments in the CPU time and memory usage for TEE when compared to REE. This is because VGG9 has a larger model size, which incurs significant complexity during encryption, decryption, and processes.

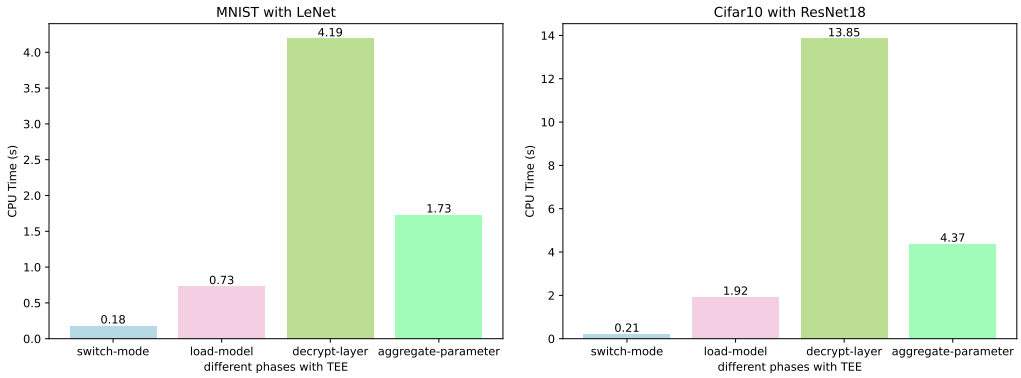
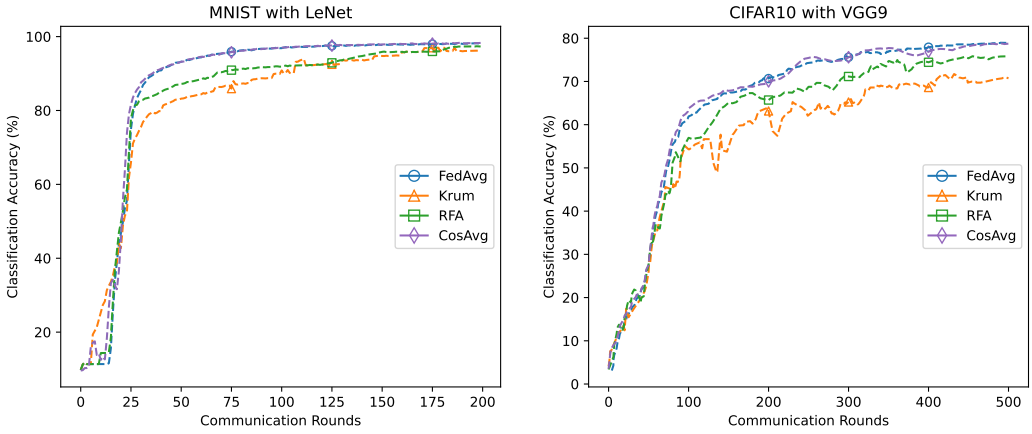


Figure 4. Time overheads with different phases with TEE.

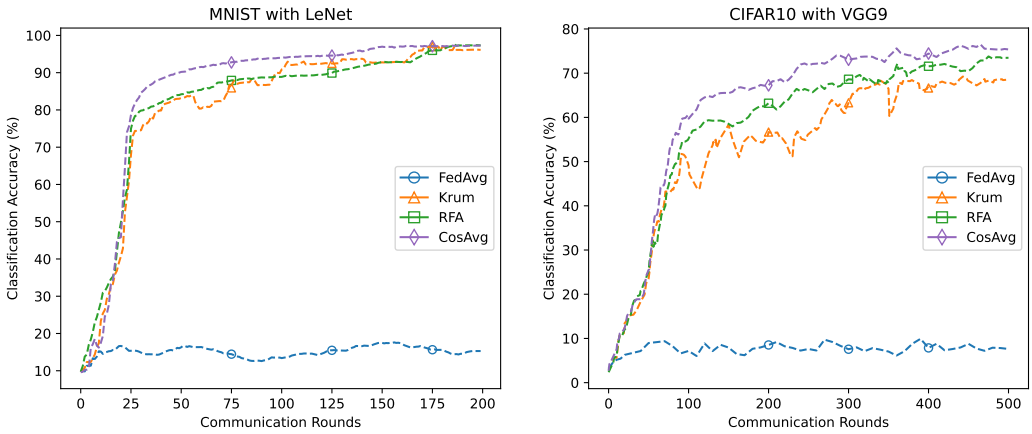
In conclusion, the TEE security mechanisms introduce relatively modest runtime overhead with smaller resource costs compared to other security frameworks. Thus, for resource-limited UAVs that prioritize security, TEE emerges as a worthwhile solution to consider.

4.3. Robustness across Various Aggregation Algorithms

We first compare the performance of the proposed CosAvg algorithm with other aggregation algorithms on two datasets, MNIST and CIFAR10. FedAvg is used as the baseline, while Krum and RFA are compared as commonly used robust aggregation algorithms. We engaged $N = 200$ clients to participate in the federated learning tasks, and the dataset was divided randomly into 200 segments. Each client independently trained for 3 epochs, locally, before transmitting the updated model parameters to the aggregation server. We set the number of FL rounds, $R1 = 200$ and $R2 = 500$, for CIFAR10. We assess the robustness of aggregation algorithms by comparing the accuracy of the aggregated models. A smaller decrease in model accuracy when malicious clients are present indicates a stronger ability to resist Byzantine attacks. Each sub-figure in Figure 5 illustrates how the accuracy of the aggregated model on the test dataset changes as the number of federated learning rounds increases under different aggregation algorithms. We assume that the attackers have complete control over the local model and can make arbitrary adjustments to the model parameters within the updates. Additionally, they possess the ability to collaborate with other malicious clients to achieve consistent updates, thereby increasing the difficulty of detection.

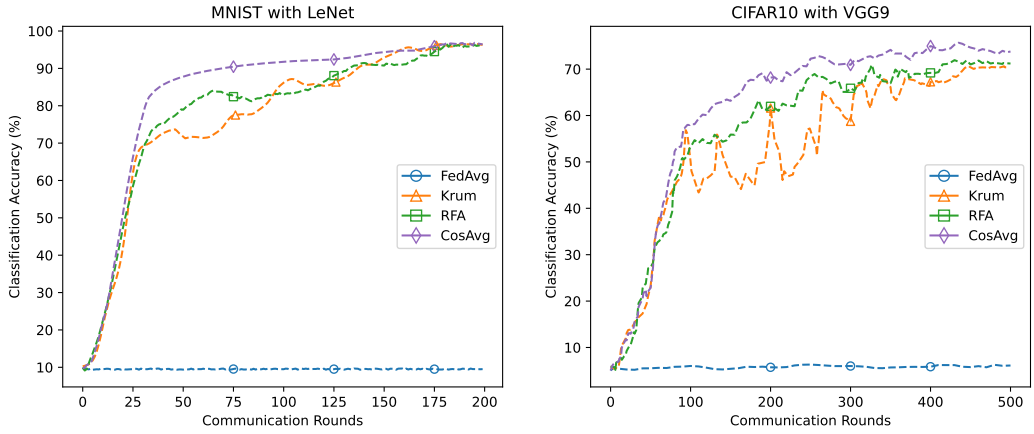


(**a**)



(**b**)

Figure 5. *Cont.*



(**c**)

Figure 5. Performance evaluation of various aggregation algorithms under both scenarios in the presence and absence of malicious clients. (**a**) Federated learning with 0% malicious clients. (**b**) Federated learning with 20% malicious clients. (**c**) Federated learning with 40% malicious clients.

When observing Figure 5 horizontally, apart from the basic FedAvg algorithm, the other aggregation algorithms exhibit a certain level of robustness. For Krum: (1) It displays significant fluctuations in prediction accuracy due to its strategy of selecting the nearest single model after each iteration. This approach amplifies the randomness and uncertainty of the aggregation outcome. (2) It demonstrates slower convergence rates and lower test accuracy compared to other algorithms. This is attributed to the fact that the model with the smallest total distance is not necessarily a benign local model. As the proportion of malicious clients increases, Krum fails to accurately differentiate between malicious and benign local model updates. In contrast, the test accuracy of RFA surpasses that of Krum because the RFA algorithm replaces the traditional average algorithm with the geometric median of single client updates. This modification grants RFA a certain degree of resilience against Byzantine attacks. The median-based approach inherently possesses greater robustness against outliers. This approach can mitigate the impact of exceptional clients to some extent; however, it cannot entirely eliminate the possibility of selecting models from abnormal clients, leading to a reduction in global model accuracy. On the other hand, CosAvg demonstrates higher prediction accuracy and faster convergence compared to the first two algorithms. It exhibits superior robustness due to its ability to recognize malicious model parameters with the measurements of cosine distances. This initial screening process helps identify exceptional clients, leaving mostly benign clients for subsequent averaging. Then the application of the average method effectively mitigates randomness and offsets errors.

When observing Figure 5 vertically, regardless of the presence or absence of malicious clients, the accuracy of the CosAvg surpasses that of the other two robustness algorithms. This achievement underscores the robust nature of our approach and its ability to effectively handle adversarial scenarios. In scenarios with malicious clients, the CosAvg showcases remarkable resilience by maintaining its high accuracy in the confront of malicious clients. The utilization of cosine similarity aggregation contributes to this robustness, allowing the algorithm to mitigate the impact of outlier updates that might originate from malicious sources. Comparatively, the other two robustness algorithms display a relatively steeper decline in accuracy in the presence of malicious clients. Furthermore, even in scenarios without malicious clients, our algorithm does not incur a significant loss in accuracy, making it a dependable choice under normal conditions. This consistency in performance

across varying scenarios substantiates the efficacy of our algorithm as a robust solution for federated learning.

5. Conclusions

In this study, we investigated runtime overhead across various secure aggregation frameworks and compared robustness across various secure aggregation algorithms in the context of UAV-assisted federated learning. From the overhead perspective, we found that the TEE security framework demonstrated a smaller resource cost compared to DP and HE, despite introducing additional computational and memory overhead due to the security mechanisms. The overhead increase in TEE was relatively modest, making it a promising security solution for resource-constrained UAVs that prioritize security. In terms of robustness, our proposed CosAvg algorithm outperformed FedAvg, Krum, and RFA. Regardless of the presence or absence of malicious clients, CosAvg maintained high accuracy, highlighting its robustness and capability to effectively handle adversarial scenarios. The TEE security framework, in conjunction with the CosAvg aggregation algorithm, presents a potent combination for secure and robust federated learning. TEE provides efficient and secure model aggregation, and CosAvg offers robustness against potential Byzantine attacks. This combination proves especially useful for UAV-assisted federated learning where the need for security and robustness is paramount.

Future prospects: The foundation of UAV security lies in TEE, yet the security foundation of TEE itself encounters certain challenges. During secure computing operations, there may also exist additional intricate threats in the TEE, such as side-channel attacks. The underlying principle of these attacks involves the extraction of incidental information generated during the system runtime through specific channels, subsequently leveraging this incidental information to infer confidential internal system data. In the future, it will be necessary to explore tailored defense mechanisms that encompass both hardware and software components against specific side-channel threat models within this system. This endeavor is significant for safeguarding the security of the UAV-assisted federated learning process.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; investigation, J.L.; resources, L.N.; data curation, J.L.; writing—original draft preparation, J.L., B.J.; writing—review and editing, J.L., B.J.; visualization, J.L.; supervision, L.N., P.Z.; project administration, L.N.; funding acquisition, L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was sponsored by the General Program of Continuous Support Foundation of Shenzhen City (no. 20220715114600001), the SZTU-CSG Cooperation Research Project (no. 20231064010043), and the Scientific Research Capacity Improvement Project from Guangdong Province (no. 2021ZDJS109).

Data Availability Statement: The datasets used in this study are publicly available. For the MNIST dataset, it can be accessed and downloaded from the official website at http://yann.lecun.com/exdb/mnist/ (accessed on 30 August 2023). For the CIFAR-10 dataset, it can be accessed and downloaded from the official website at http://www.cs.toronto.edu/~kriz/cifar.html (accessed on 30 August 2023).

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest related to this work.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	unmanned aerial vehicle
MEC	mobile edge computing
TEE	trusted execution environment
MPC	secure multi-party computation
DP	differential privacy
HE	homomorphic encryption

References

1. Michailidis, E.T.; Maliatsos, K.; Skoutas, D.N.; Vouyioukas, D.; Skianis, C. Secure UAV-aided mobile edge computing for IoT: A review. *IEEE Access* **2022**, *10*, 86353–86383. [CrossRef]
2. Liu, Z.; Cao, Y.; Gao, P.; Hua, X.; Zhang, D.; Jiang, T. Multi-UAV network assisted intelligent edge computing: Challenges and opportunities. *China Commun.* **2022**, *19*, 258–278. [CrossRef]
3. Mekdad, Y.; Aris, A.; Babun, L.; El Fergougui, A.; Conti, M.; Lazzeretti, R.; Uluagac, A.S. A survey on security and privacy issues of UAVs. *Comput. Netw.* **2023**, *224*, 109626. [CrossRef]
4. Wang, Y.; Su, Z.; Zhang, N.; Benslimane, A. Learning in the air: Secure federated learning for UAV-assisted crowdsensing. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 1055–1069. [CrossRef]
5. Zhong, X.; Yuan, X.; Yang, H.; Zhong, C. UAV-assisted hierarchical aggregation for over-the-air federated learning. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 807–812.
6. Qu, Y.; Dai, H.; Zhuang, Y.; Chen, J.; Dong, C.; Wu, F.; Guo, S. Decentralized federated learning for UAV networks: Architecture, challenges, and opportunities. *IEEE Netw.* **2021**, *35*, 156–162. [CrossRef]
7. Song, Z.; Qin, X.; Hao, Y.; Hou, T.; Wang, J.; Sun, X. A comprehensive survey on aerial mobile edge computing: Challenges, state-of-the-art, and future directions. *Comput. Commun.* **2022**, *191*, 233–256. [CrossRef]
8. Lim, W.Y.B.; Huang, J.; Xiong, Z.; Kang, J.; Niyato, D.; Hua, X.S.; Leung, C.; Miao, C. Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5140–5154. [CrossRef]
9. Pham, Q.V.; Le, M.; Huynh-The, T.; Han, Z.; Hwang, W.J. Energy-efficient federated learning over UAV-enabled wireless powered communications. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4977–4990. [CrossRef]
10. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep models under the GAN: Information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 603–618.
11. Roszel, M.; Norvill, R.; State, R. An Analysis of Byzantine-Tolerant Aggregation Mechanisms on Model Poisoning in Federated Learning. In Proceedings of the International Conference on Modeling Decisions for Artificial Intelligence, Sant Cugat, Spain, 30 August–2 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 143–155.
12. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
13. Phong, L.T.; Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1333–1345. [CrossRef]
14. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
15. Geyer, R.C.; Klein, T.; Nabi, M. Differentially private federated learning: A client level perspective. *arXiv* **2017**, arXiv:1712.07557.
16. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 118–128 .
17. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust Aggregation for Federated Learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1142–1154. [CrossRef]
18. Messaoud, A.A.; Mokhtar, S.B.; Nitu, V.; Schiavoni, V. Shielding federated learning systems against inference attacks with ARM TrustZone. In Proceedings of the 23rd ACM/IFIP International Middleware Conference, Quebec, QC, Canada, 7–11 November 2022; pp. 335–348.
19. Kuznetsov, E.; Chen, Y.; Zhao, M. Securefl: Privacy preserving federated learning with sgx and TrustZone. In Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 14–17 December 2021; pp. 55–67.
20. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. In *Concurrency: The Works of Leslie Lamport*; Association for Computing Machinery: New York, NY, USA 2019 ; pp. 203–226.
21. Suzaki, K.; Nakajima, K.; Oi, T.; Tsukamoto, A. Ts-perf: General performance measurement of trusted execution environment and rich execution environment on intel sgx, arm TrustZone, and risc-v keystone. *IEEE Access* **2021**, *9*, 133520–133530. [CrossRef]
22. Alves, T.; Felton, D. TrustZone: Integrated Hardware and Software Security. *Inf. Q.* **2004**, *3*, 18–24.

23. Liu, R.; Srivastava, M. Protc: Protecting drone's peripherals through arm TrustZone. In Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, Niagara Falls, NY, USA, 23 June 2017; pp. 1–6.
24. Zhang, H.; Hanzo, L. Federated learning assisted multi-UAV networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14104–14109. [CrossRef]
25. Mo, F.; Haddadi, H.; Katevas, K.; Marin, E.; Perino, D.; Kourtellis, N. PPFL: Privacy-Preserving Federated Learning with Trusted Execution Environments. In Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys'21, Virtual Event, 24 June–2 July 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 94–108.
26. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
27. Li, Z.; Zhang, J.; Liu, L.; Liu, J. Auditing Privacy Defenses in Federated Learning via Generative Gradient Leakage. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 10132–10142.
28. Wang, Y.; Deng, J.; Guo, D.; Wang, C.; Meng, X.; Liu, H.; Ding, C.; Rajasekaran, S. Sapag: A self-adaptive privacy attack from gradients. *arXiv* **2020**, arXiv:2009.06228.
29. Zhu, J.; Blaschko, M.B. R-GAP: Recursive Gradient Attack on Privacy. *arXiv* **2020**, arXiv:2010.07733.
30. Jin, X.; Chen, P.Y.; Hsu, C.Y.; Yu, C.M.; Chen, T. CAFE: Catastrophic Data Leakage in Vertical Federated Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J.W., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 994–1006.
31. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 20), Online, 15–17 July 2020; pp. 493–506.
32. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Vincent Poor, H. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
33. McMahan, H.B.; Ramage, D.; Talwar, K.; Zhang, L. Learning Differentially Private Language Models Without Losing Accuracy. *arXiv* **2017**, arXiv:1710.06963.
34. Zhang, X.; Gu, H.; Fan, L.; Chen, K.; Yang, Q. No Free Lunch Theorem for Security and Utility in Federated Learning. *ACM Trans. Intell. Syst. Technol.* **2022**, *14*, 1–35. [CrossRef]
35. Shi, J.; Wan, W.; Hu, S.; Lu, J.; Yu Zhang, L. Challenges and Approaches for Mitigating Byzantine Attacks in Federated Learning. In Proceedings of the 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Wuhan, China, 9–11 December 2022; pp. 139–146.
36. Fang, M.; Cao, X.; Jia, J.; Gong, N. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Boston, MA, USA, 12–14 August 2020; pp. 1605–1622.
37. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; Singh, A., Zhu, J., Eds.; Volume 54 , pp. 1273–1282.
38. El Mhamdi, E.M.; Guerraoui, R.; Rouault, S. The Hidden Vulnerability of Distributed Learning in Byzantium. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; Volume 80, pp. 3521–3530.
39. Fung, C.; Yoon, C.J.; Beschastnikh, I. The limitations of federated learning in sybil settings. In Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), San Sebastian, Spain, 14–15 October 2020; pp. 301–316.
40. Meng, Z.; Xia, X.; Xu, R.; Liu, W.; Ma, J. HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR. *IEEE Trans. Intell. Veh.* **2023** , 1–13. [CrossRef]
41. Xia, X.; Bhatt, N.P.; Khajepour, A.; Hashemi, E. Integrated Inertial-LiDAR-Based Map Matching Localization for Varying Environments. *IEEE Trans. Intell. Veh.* **2023** , 1–12. [CrossRef]
42. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.