Standing Firm in 5G: A Single-Round, Dropout-Resilient Secure Aggregation for Federated Learning

Yiwei Zhang
Purdue University
West Lafayette, IN, USA
yiweizhang@purdue.edu

Rouzbeh Behnia
University of South Florida
Tampa, FL, USA
behnia@usf.edu

Imtiaz Karim
Purdue University
West Lafayette, IN, USA
karim7@purdue.edu

Attila A. Yavuz
University of South Florida
Tampa, FL, USA
attilaayavuz@usf.edu

Elisa Bertino
Purdue University
West Lafayette, IN, USA
bertino@purdue.edu

Abstract

Federated learning (FL) is well-suited to 5G networks, where many mobile devices generate sensitive edge data. Secure aggregation protocols enhance privacy in FL by ensuring that individual user updates reveal no information about the underlying client data. However, the dynamic and large-scale nature of 5G-marked by high mobility and frequent dropouts-poses significant challenges to the effective adoption of these protocols. Existing protocols often require multi-round communication or rely on fixed infrastructure, limiting their practicality. We propose a lightweight, single-round secure aggregation protocol designed for 5G environments. By leveraging base stations for assisted computation and incorporating precomputation, key-homomorphic pseudorandom functions, and t-out-of-k secret sharing, our protocol ensures efficiency, robustness, and privacy. Experiments show strong security guarantees and significant gains in communication and computation efficiency, making the approach well-suited for real-world 5G FL deployments.

CCS Concepts

• **Security and privacy** → **Security protocols**; *Privacy protections*.

Keywords

5G; Federated Learning; resilient networks; privacy

1 Introduction

Federated learning (FL) has emerged as a powerful approach for enabling privacy-preserving machine learning across distributed devices, allowing users to collaboratively train models without sharing raw data [19]. While FL has been deployed over conventional communication infrastructures, the rapid rollout of fifth-generation (5G) networks introduces both opportunities and new challenges. On the one hand, 5G offers high bandwidth, low latency, and support for massive device connectivity—features that make it well-suited for large-scale, on-device FL deployments. Real-world applications already highlight FL's potential in 5G settings, such as real-time analytics for connected vehicles and privacy-preserving medical diagnostics [17, 21]. FL can also benefit network providers by improving tasks like channel estimation [15], thereby enhancing overall network performance while maintaining user privacy. On the other hand, these same 5G features create a highly dynamic environment [13]. Devices frequently join and leave the network, connectivity conditions fluctuate, and the scale of participation can surpass what existing FL protocols were designed to handle.

These 5G-specific conditions pose significant obstacles for secure aggregation [6], a critical component of FL that ensures individual model updates remain private during server-side aggregation. Most existing secure aggregation techniques assume relatively stable device participation and moderate network scale, often relying on multiple rounds of interaction or fixed infrastructure. In contrast, a practical secure aggregation protocol for 5G must operate efficiently under high churn, large user populations, and minimal communication rounds-all while preserving strong security guarantees.

Requirements of 5G for FL. 5G's capacity to connect large numbers of devices simultaneously greatly expands the pool of potential FL participants, but it also introduces significant communication and computation overhead if aggregation involves multiple rounds of interaction. Meanwhile, 5G networks exhibit high mobility and signaling, leading to unpredictable connectivity and higher churn rates, which can degrade the reliability and speed of FL aggregation. Therefore, secure aggregation solutions in 5G must satisfy several key requirements: 1) **Scalability.** Secure aggregation introduces additional computation and communication overhead in addition to the local training on mobile devices. Therefore, given the resource constraints of these battery-powered devices, this can directly impact the performance and scalability of these systems. Since each user's training parameters must be collected, the overhead of computation and communication should remain manageable even when thousands or millions of devices are involved. 2) **Resilience.** Given the dynamic nature of 5G, where user devices (e.g., cell phones) and even the base stations can go offline unexpectedly, the secure

aggregation mechanism must be robust against dropouts. 3) **Compatibility.** To enable seamless deployment, the protocol should be compatible with existing 5G infrastructures without demanding substantial hardware or protocol modifications.

Current Secure Aggregation Solutions. Existing approaches to secure aggregation primarily rely on secure multi-party computation (MPC) [12, 18] or pairwise masking [4, 5, 14]. For instance, Flamingo [18] employs a multi-round protocol where users reuse a secret to generate masks; a subset (decryptors) then interacts over multiple rounds to reveal active user masks and eliminate offline user contributions. e-SeaFL [4], based on pairwise masking, reduces the communication overhead by introducing assisting nodes that hold shared secrets, achieving single-round secure aggregation. OPA [14] also follows the pairwise masking paradigm, but instead of relying on pre-shared secrets, it uses key-homomorphic cryptography. Users interact with a committee to remove global masks by exchanging fresh auxiliary information in each training round.

However, these methods face limitations under 5G conditions. While Flamingo can handle dropouts via secret sharing, its multi-round interactions significantly increase communication overhead. e-SeaFL relies on a designated set of assisting nodes, and if these nodes go offline unexpectedly, it can cause the entire protocol to fail. Both protocols have stringent requirements and rely on complex public key infrastructures (PKIs), making their deployment infeasible in fast-evolving 5G networks. OPA uses a stateless, one-shot design without pre-shared secrets, running only one training round per execution. To support multiple FL rounds, it must restart with new auxiliary exchanges each round, incurring high overhead. It also lacks precomputation mechanisms to alleviate the computational and communication overhead of its single-round approach, making it less suitable for resource-constrained 5G environments.

Our Contributions. To address these challenges, we propose a novel secure aggregation framework tailored to FL in large-scale 5G environments. Our approach leverages base stations to assist the server in aggregating user updates to significantly reduce the overall communication and computation overhead, achieving a *single-round* secure aggregation protocol. We also introduce a simple but effective pre-computation method to meet the stringent performance requirements of mobile devices in 5G networks. By incorporating key-homomorphic pseudo-random functions (KHPRFs) and a robust t-out-of-k secret sharing scheme, our protocol tolerates both user equipment and base station dropouts without compromising security or correctness. Crucially, our solution integrates seamlessly with standard 5G architectures and does not impose significant additional infrastructure requirements. In summary, our main contributions include:

- A *single-round* secure aggregation protocol optimized for large-scale 5G settings, supported by *precomputation* techniques and base stations to minimize computation and communication overhead and the risk of data leakage.
- Resilience mechanisms that ensure continuous operation despite unexpected device or base station dropout, thus maintaining reliable model convergence in 5G deployments.
- A thorough security analysis and experimental evaluation showing the computational and communication efficiency of our approach, as well as its robust security guarantees.

2 Preliminaries and Models

Following [5, 18], we adopt the definitions of privacy for secure aggregation protocols as follows.

Definition 2.1 (Key-Homomorphic Pseudorandom Function (KH-PRF) [7]). A key-homomorphic pseudorandom function is KHPRF : $\mathcal{K} \times \mathcal{M} \to \mathcal{Y}$, where $\mathcal{K}$ is the key space, $\mathcal{M}$ is the domain of inputs, and $\mathcal{Y}$ is the output space. This function satisfies: 1) *Key-Homomorphism*: For any keys $k_1, k_2 \in \mathcal{K}$ and any input $m \in \mathcal{M}$, $\text{KHPRF}(k_1 + k_2, m) = \text{KHPRF}(k_1, m) + \text{KHPRF}(k_2, m)$ where the addition operations are performed in the appropriate key and output groups. This property naturally extends to any finite sums of keys, i.e., $\text{KHPRF}(\sum_{i \in I} k_i, m) = \sum_{i \in I} \text{KHPRF}(k_i, m)$, for any index set $I \subseteq \{1, \ldots, n\}$. 2) *Pseudorandomness*: For a uniformly chosen $k \in \mathcal{K}$, the function $\text{KHPRF}(k, \cdot)$ is computationally indistinguishable from a truly random function mapping $\mathcal{M}$ to $\mathcal{Y}$.

Definition 2.2 (t-out-of-n Secret Sharing [22]). A secret sharing scheme $SS = \{SS.\text{Split}, SS.\text{Recover}\}$ for threshold t out of n operates over a message space $\mathcal{M}$. It consists of two algorithms:

- $\{s_1, \ldots, s_n\} \leftarrow SS.\text{Split}(s)$: Given a secret $s \in \mathcal{M}$, the algorithm outputs n shares $\{s_1, \ldots, s_n\}$. Any subset of at least t shares is sufficient to recover s, whereas any subset of fewer than t shares reveals no information about s.
- $s \leftarrow SS.\text{Recover}(S')$: Given any subset $S' \subseteq \{s_1, \ldots, s_n\}$ of size $|S'| \geq t$, the algorithm reconstructs and outputs the secret s. If $|S'| < t$, it outputs $\perp$ (failure).

Definition 2.3 (α-summation ideal functionality [5]). Given p, n, d as integers and consider a set $L \subseteq [n]$ with associated data vectors $\mathcal{W}_L := \{w_i\}_{i \in L}$ where $w_i \in \mathbb{Z}_p^d$. Given a threshold $0 \leq \alpha \leq 1$ and Q_L as the set of partitions of L and a set of pairwise disjoint subsets $\{L_1, \ldots, L_l\} \in Q_L$, the α-summation ideal functionality $\mathcal{F}_{w,\alpha}(\cdot)$ computes $\mathcal{F}_{w,\alpha}(\{L_i\}_{i \in [1,\ldots,l]}) \to \{s_i\}_{i \in [1,\ldots,l]}$, where

$$\forall j \in [1, \ldots, l], s_j = \begin{cases} \sum_{j \in Q_L} w_j & \text{if } Q_L| \geq \alpha|L| \\ \perp & \text{else.} \end{cases}$$

Definition 2.4 (Privacy of Secure Aggregation [5]). Let Σ be a Shamir secret sharing protocol instantiated with a security parameter κ, and $\mathcal{F}$ be an ideal functionality that only outputs the aggregation if at least αn defined as honest users are participating, where $0 < \alpha \leq 1$. An aggregation protocol Γ is said to preserve privacy against an adversary $\mathcal{A}$ if there exists a probabilistic polynomial-time (PPT) simulator Sim such that, for any iteration $t \in [T]$ and any set of input vectors $\mathcal{W}^t = \{w_1, \ldots, w_n\}$, the output produced by Sim is computationally indistinguishable from the adversary's view. The adversary $\mathcal{A}$ is assumed to have control over NC, a λ_P fraction of users U_c, and a λ_A fraction of assisting nodes BS_c. Its view comprises the combined view of the compromised server NC^*, the set of compromised users, and the set of compromised assisting nodes: $\text{Real}^\Gamma(\text{Adv}, \{w_i\}_{i \notin U_c}) \approx_\kappa \text{Simul}^{\Gamma, \mathcal{F}_{\mathcal{W}^t, \alpha}(\cdot)_{\{w_i \notin U_c\}}}(\text{Adv})$.

2.1 System Model

5G-Integrated FL Ecosystem. As illustrated in Figure 1, our architecture comprises three core components: *user equipment (UE)*, *base stations (BSs)*, and the *core network (CN)*. It harnesses 5G's high bandwidth, low latency, edge-processing capabilities as well
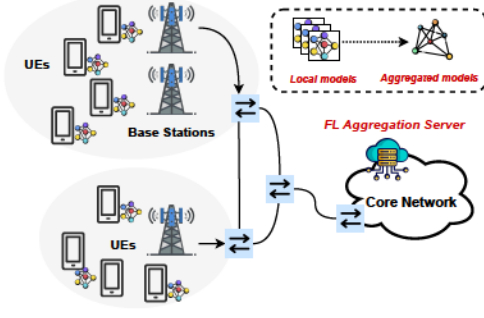
Figure 1: 5G+FL Framework.

as security features to support efficient, privacy-preserving model training across diverse user devices.

UEs (e.g., mobile or IoT devices) collect local data and train models at the network edge. They typically possess basic edge-computing capabilities, allowing them to perform local learning without uploading raw data to the aggregation server and masking their local updates before transmission. Each UE operates under a BS for connectivity; BSs often possess edge-computing resources to help with partial unmasking and handling potential device dropouts. The CN orchestrates system-wide coordination and security. Typically, the *Authentication Server Function* (AUSF) in the CN ensures secure device authentication, while the *Application Function* (AF), either hosted in the CN or provided by an external third party, delivers user-facing services. When integrating with FL functionality, a dedicated AF module (AF_{FL}) coordinates the end-to-end FL process by collecting UE updates, aggregating them into a global model, and distributing the result to participating UEs. Robust end-to-end encryption and integrity checks among the three components are enforced via the 5G-AKA protocol [1].

FL Workflow. The FL procedure comprises two main phases. In the one-time *Setup Phase*, cryptographic materials are configured for UEs and BSs, including establishing secret shares to protect local updates and tolerate BS dropouts. In the iterative *Aggregation Phase*, UEs independently train local models, mask their updates, and send them to AF_{FL}. The AF_{FL} then unmasks and aggregates these updates to compute a global model, which is subsequently distributed back to the UEs for further local training.

2.2 Threat Model

Based on the 3GPP Technical Specifications [1–3] and relevant literature [17, 24], we consider the following adversarial behaviors and practical constraints within our 5G-based FL framework:

- We assume that the CN in 5G is generally trusted, with the exception of the AF_{FL} responsible for model aggregation in FL. Since AF_{FL} may be provided by a third-party service, it can behave maliciously, attempting to infer private user data or conducting membership inference attacks based on aggregated updates [9, 10, 23]. In contrast, other CN components (e.g., AUSF) are considered benign, as they typically reside within the trusted 5G Core Network infrastructure [1].

- UEs may drop out during FL training due to network instability, node mobility, or active adversarial efforts [16, 20]. Their unpredictable mobility across different BSs can further complicate model consistency. Likewise, BSs are susceptible to outages or unavailability caused by network failures, targeted attacks, or other system constraints [13].

- To leverage the existing 5G security mechanisms [2] rather than introducing new ones (such as PKI), we consider that the FL process begins only after the 5G-AKA procedure is completed. This ensures that communication keys (e.g., long-term keys and session keys) have been securely negotiated and distributed, thus encrypting and protecting the integrity of all transmitted information among UEs, BSs, and CN (AUSF and AF_{FL}). Consequently, even if adversaries intercept communication channels, they cannot decrypt or forge valid messages unless they obtain direct access to a compromised user device[2].

3 Our Protocol

We now describe our 5G-based FL secure aggregation protocol (see Figure 2). Our protocol is designed to meet the high connectivity, mobility, resilience and compatibility demands of 5G. By combining threshold secret sharing, KHPRFs, and BS-assisted unmasking, it protects individual user data from inference attacks and remains robust against UE and BS dropouts. Each UE and BS has single-round communication with the CN, maintaining high efficiency.

Our system consists of three primary entities: 1) n UEs (denoted by U_i where $i \in \{1, ..., n\}$); 2) k BSs (denoted by BS_j where $j \in \{1, ..., k\}$) in a certain region; and 3) a single CN that hosts an AF as the FL aggregation server (AF_{FL}) as well as an AUSF to forward sensitive messages. The protocol operates in two main phases: an initial *Setup* phase (one-time) to establish cryptographic secrets, followed by a single-round *Aggregation* phase (iterative) for secure model updates.

3.1 Setup Phase

The Setup phase is executed once to bootstrap trust and secrets among the UEs and the BSs, ensuring resilience against BS dropouts and enabling secure masked aggregation.

Step 1.1: UE secret generation and distribution. Each UE initializes a secret S^{U_i}. To facilitate the recovery of user masks during aggregation, this secret S^{U_i} is partitioned into k splits, $\{S_j^{U_i}, \forall j \in [k]\}$, using a t-out-of-k secret sharing scheme $SS.\text{Split}(\cdot)$. The UE sends these shares to the CN via its currently assigned BS. The AUSF within CN then forwards and distributes one share to each BS in the designated region. This ensures that the UE secret can be reconstructed if at least t shares are collected, thereby mitigating the impact of BS dropouts.

3.2 Aggregation Phase

At each iteration t, the protocol executes three steps to securely aggregate local updates and refresh the global model.

Step 2.1: Local model training and masked update generation. Each U_i trains a local model on its private dataset and obtains a

---

[1]We note that we can lift these trust assumptions on CN and AUSF by assuming a simple digital signature scheme akin to the malicious model in [4].

[2]Similar to previous work on secure aggregation [4, 18] we do not consider impersonated or compromised UEs and BSs as our paper focuses on secure aggregation instead of data poisoning.
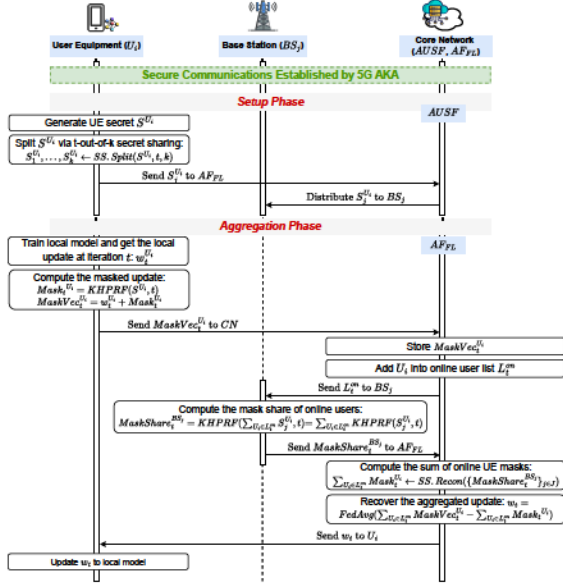
Figure 2: 5G-based FL secure aggregation protocol.

parameter update[3] $\mathbf{w}_t^{U_i}$. To protect $\mathbf{w}_t^{U_i}$ from direct exposure to the AF_{FL}, U_i computes a mask vector using the KHPRF: $Mask_t^{U_i} = KHPRF(S^{U_i}, t)$. The final masked update is $MaskVec_t^{U_i} = \mathbf{w}_t^{U_i} + Mask_t^{U_i}$, which is then sent to to the AF_{FL}.

Since the mask is derived solely from S^{U_i} and the known iteration index t, our protocol adopts a **pre-computation** strategy. That is, a UE can precompute all necessary masks (such as T masks for the first T iterations) before the runtime aggregation phase, minimizing on-the-fly computation costs.

Step 2.2: Online user registration and mask share computation. Once the AF_{FL} receives $MaskVec_t^{U_i}$, it logs U_i in the *online UE list* L_t^{on}, identifying UEs that remain active for iteration t. The AF_{FL} checks that $|L_t^{on}| \geq \alpha n$; if so, it shares this list with all BSs (in a certain region).

Each BS_j computes its *mask share* by summing the secret shares received from all UEs in L_t^{on} and applying the KHPRF: $MaskShare_t^{BS_j} = \text{KHPRF}(\sum_{U_i \in L_t^{on}} S_j^{U_i}, t) = \sum_{U_i \in L_t^{on}} \text{KHPRF}(S_j^{U_i}, t)$. Owing to the key-homomorphic property, the final output is equivalent to the sum of individual $KHPRF(\cdot)$ evaluations, enabling the AF_{FL} to only recover the *aggregated mask* while never seeing any single user's update and mask.

Step 2.3: Secure aggregation and global model update. Using the masked UE updates and the mask shares, the AF_{FL} first reconstructs the aggregated mask of online UE masks via the t-out-of-k secret reconstruction algorithm: $\sum_{U_i \in L_t^{on}} Mask_t^{U_i} = SS.\text{Recover}(\{MaskShare_t^{BS_j}\}_{j \in J})$, where J is a subset of BSs which are online.

The AF_{FL} then computes the global aggregated update $\mathbf{w}_t$ by subtracting the sum of masks from the sum of all masked UE updates, then applying a standard FL aggregator (e.g., FedAvg [19]): $\mathbf{w}_t = \text{FedAvg}(\sum_{U_i \in L_t^{on}} MaskVec_t^{U_i} - \sum_{U_i \in L_t^{on}} Mask_t^{U_i})$. This sum

---

[3]The protocol accommodates both gradient- and parameter-based FL variants.

of masks can make sure to "unmask" the global update without revealing individual user parameters, i.e., only the aggregated result $\mathbf{w}_t$ is revealed, preventing the AF_{FL} from accessing any user-level update directly.

After computing the global update $\mathbf{w}_t$, the AF_{FL} then propagates it to all participating UEs. UEs then incorporate these aggregated parameters into their local models, completing the t-th iteration. Notably, new or reconnected UEs can join seamlessly after completing their own setup phase, while the aggregation phase continues iteratively until the global model converges or meets predefined performance criteria.

3.3 Security Analysis

THEOREM 3.1. *The protocol proposed above with n users, k base stations and the aggregation server AF_{FL} is private against an adversary $\mathcal{A}$ where $\mathcal{A}$ is able to compromise $(1 - \alpha)n$ users, $k - 1$ base stations and the aggregation server AF_{FL}.*

PROOF. Following Definition 2.4, we consider a simulator C and establish indistinguishability through a standard hybrid argument. This is done by presenting a set of successive hybrids that are computationally indistinguishable. We start by defining the behavior of C during the Setup and Aggregation phases. In the Setup phase, honest users U_i compute their secret shares $\{S_1^{U_i} \ldots, S_k^{U_i}\}$ and encrypt them (using AKA-ENC) for the target BS. The encrypted shares are distributed to the base stations via AUSF. During the aggregation phase, users compute their masked update $MaskVec_t^{U_i}$ and send it to AF_{FL}. Next, AF_{FL} adds the user to an authenticated list of participating users. We now present our hybrids.

Hyb0: This hybrid represents the actual execution of the protocol, during which $\mathcal{A}$ interacts with the honest entities.

Hyb1: C is introduced. C is assumed to possess all the secrets of the honest parties.

Hyb2: The behavior of the honest parties U_i and base stations BS_j is modified by selecting a random shared secret key from the key space $\mathcal{K}$ instead of executing $SS.\text{Split}(\cdot)$ algorithm. The security of the secret sharing protocol (e.g., as in [5, 22]) and the encryption scheme ensure the indistinguishability of this hybrid from the previous one.

Hyb3: Each honest user U_i replaces the masked update $MaskVec_t^{U_i}$ sent to AF_{FL} with a random vector $MaskVec_t'^{U_i}$. Given our protocol requires at least one base station to be honest, the indistinguishability of this hybrid is ensured, as $\mathcal{A}$ does not have knowledge of at least one honest base station. Therefore, in Real, the masked update follows the same distribution as $MaskVec_t'^{U_i}$ in Simul.

Hyb4: The aggregated masking term $MaskShare_t^{BS_j}$ outputted by the honest base stations BS_j is replaced with $MaskShare_t'^{BS_j}$ by using the ideal functionality $\mathcal{F}_{\mathbf{w}, \alpha}(L_t'^{on})$ where $L_t'^{on}$ is the list of honest participating users. Since $\mathcal{A}$ has no knowledge of the honest entities' shared secret, the distribution of the aggregated masking term in Simul is identical to that in Real and the view of this hybrid remains indistinguishable.

Hyb5: C sends the output of the ideal functionality as the intermediate model $\mathbf{w}_t$. Note that the ideal functionality does not return $\perp$, given the condition on the fraction of honest users. Consequently, this hybrid remains indistinguishable from the previous one, as the local updates are unknown to $\mathcal{A}$ in Real.

From the above, we have demonstrated that the view of all corrupted parties under the control of $\mathcal{A}$ is computationally indistinguishable from their view in Real. □

4 Evaluation

We evaluate our 5G-based FL secure aggregation protocol by comparing it against state-of-the-art alternatives. The goal is to demonstrate the efficiency, communication overhead, and resilience of our approach in realistic 5G network scenarios.

4.1 Experiment Setup

Implementation We implemented our protocol in Python (approximately 1,200 lines of code) using the ABIDES [8] simulation framework, which enables controlled testing of multi-round FL aggregation protocols, allowing us to replicate realistic 5G network behaviors in a simulated environment. We used ASCON [11] as the pseudo-random function (PRF) for mask generation, given its lightweight design and suitability for resource-constrained devices.

Experiment Environments. All experiments were conducted on an x86_64 Linux server equipped with an AMD Ryzen Threadripper PRO 5965WX (24-core CPU), 256 GB of RAM, and three NVIDIA GeForce RTX 3090 GPUs for model training. We do not include model-training overhead in our measurements, as our primary focus is the communication and computation costs introduced by the secure aggregation protocol itself.

Evaluation Metrics. We assess our protocol using two primary metrics. *Efficiency* is quantified by computation time and communication overhead during both the setup and aggregation phases. *Resilience* is measured by the ability of the protocol to maintain global model accuracy in the presence of UE and BS dropouts.

Baselines. We compare our protocol with two secure aggregation protocols, e-SeaFL and Flamingo, as they have a certain ability to be resistant to device dropout. Under similar capabilities, we record the computation time and communication overhead to evaluate the efficiency of our protocol. For resilience evaluation, we include a baseline variant of our protocol that simply halts FL updates (i.e., distribute the original global model update) if the aggregated update cannot be correctly reconstructed.

Parameter Selection. For a fair comparison, all protocols are evaluated under the same conditions. Unless otherwise stated, our experiments use four BSs ($k = 4$), and eight UEs ($n = 8$). The dropout rate is set to $\alpha = \frac{1}{3}$, meaning up to one-third of the UEs and BSs may disconnect during training. A (t, k) threshold with $t = (1 - \alpha) \cdot k$ (i.e., $t = 3$), ensuring successful reconstruction of the aggregated mask as long as at least three BSs remain online. Moreover, we consider simulated dropout scenarios by reducing the number of UEs from 8 to 2 and BSs from 4 to 2, running tests over 10 training iterations. The key evaluation metric is the accuracy of the global model, which measures the impact of dropouts on FL performance, allowing us to analyze the robustness of the protocol under varying dropout conditions.

4.2 Results

4.2.1 Efficiency. Figure 3 reports the computation and communication overhead of our solution compared with e-SeaFL and Flamingo.
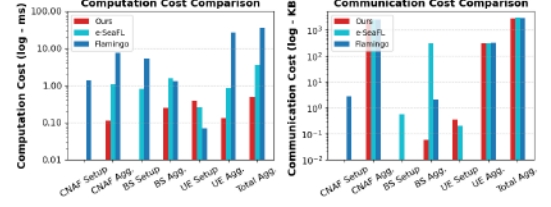

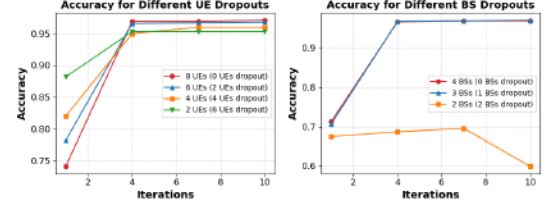**Figure 3: Efficiency Comparison.**

Figure 4: Dropout Resilience.

Overall, our protocol demonstrates significantly lower aggregation-phase latency and competitive communication costs.

Setup phase. During the one-time setup, our protocol imposes minimal load on both the AF_{FL} and the BSs, as most computational tasks are offloaded to the UEs (e.g., precomputation). Although this leads to a slightly higher setup cost on the UE side compared to e-SeaFL and Flamingo, the precomputation of masks at this phase substantially reduces overheads during aggregation. In contrast, Flamingo and e-SeaFL have to perform distributed key generation or shared secret computations among the BSs and the AF_{FL}, which increase the setup cost on those devices.

Aggregation phase. Once FL procedure has been initialized, our protocol achieves markedly faster secure aggregation than e-SeaFL and Flamingo. This improvement is primarily due to: (i) shifting the bulk of cryptographic tasks to a one-time precomputation step in the setup phase; and (ii) employing a key-homomorphic framework that avoids per-BS decryption operations when handling dropped or offline clients. In comparison, Flamingo tends to incur significant additional steps to manage these dropouts at the BS level (such as decryption), while e-SeaFL, without a precomputation step, requires each BS to compute large mask vectors at each iteration.

In terms of communication cost, our protocol remains on par with or improves with respect to e-SeaFL and Flamingo. On the UE and the AF_{FL} side, our protocol has a similar communication throughput of e-SeaFL, due to the masked UE updates and aggregated final update, as well as an online user list in our protocol. Notably, BSs in our scheme benefit from transmitting far fewer bits, since each device only sends an aggregated mask share rather than a full vector. The mask shares from the BSs can be expanded on the AF_{FL}. Flamingo incurs more communication cost due to its multiple interactions between AF_{FL} and BSs for dropout management, like online UE/BS lists and encryption/decryption results, which create additional communication steps that our approach avoids.

4.2.2 Resilience to UE and BS Dropout. Given the distributed nature of FL, real-world deployments often encounter scenarios where

participating nodes may become temporarily unavailable due to network failures, device disconnections, or power constraints. Figure 4 illustrates the impact of UE and BS dropouts on the global model accuracy.

The left panel of Figure 4 shows the impact of varying UE dropouts on the global model accuracy. We can observe that even as the number of dropped UEs increases, the overall accuracy remains relatively stable. This resilience is due to the effective reconstruction of aggregated updates via active BSs. However, when a larger fraction of UEs drop out, the final accuracy experiences a slight decline due to the reduction in the amount of data contributing to the model update, which also violates the privacy guarantee 2.3.

The right panel shows the impact of BS dropouts on accuracy. When up to one BS drops out, the model maintains a high accuracy level. However, when two BSs drop out, the accuracy decreases significantly. This is expected, as our $(3, 4)$ secret sharing scheme in the experiment requires at least three BSs to reconstruct the aggregated mask correctly. With only two BSs remaining, the server is unable to aggregate UE contributions effectively, preventing meaningful global model updates and leading to a stagnation in accuracy at a lower level.

4.2.3 Summary of Findings. Our evaluation demonstrates that the proposed protocol achieves lower aggregation overhead by shifting intensive cryptographic operations to the setup phase, resulting in reduced latency compared to similar schemes. It also maintains competitive communication cost by eliminating the need for extra communication rounds common in multi-phase decryption protocols, keeping bandwidth usage comparable to or below existing methods. Finally, the use of threshold secret sharing and BS-assisted unmasking provides robust handling of moderate UE and BS dropouts, though overall reliability depends on having a sufficient number of active base stations.

5 Conclusion and Future Work

In this paper, we presented a secure aggregation framework for federated learning over 5G networks. By incorporating a single-round protocol with t-out-of-k secret sharing and key-homomorphic pseudorandom functions, our approach efficiently safeguards user data while maintaining reliable model updates, even in dynamic and large-scale 5G environments. The empirical results confirm the protocol's ability to preserve privacy, tolerate adversarial conditions, and adapt to core characteristics of future wireless ecosystems.

Looking ahead, several promising research directions could further enhance this framework. First, an adaptive base station selection mechanism, potentially supported by machine learning, can better handle user mobility by dynamically choosing the most appropriate base stations. Second, strategies for share reassignment and revocation may strengthen security, ensuring continuous protection when the network topology changes. Third, more advanced or flexible threshold secret sharing schemes-including the integration of zero-knowledge proofs-may bolster resilience against adversarial threats and fluctuating network conditions. Last, evaluating the protocol on real-world devices remains an important next step, so we leave this as future work and consider it a key step toward practical deployment.

References

[1] 3GPP. 2020. *Study on authentication and key management for applications based on 3GPP credential in 5G.* Technical Specification.

[2] 3GPP. 2024. *Authentication and Key Management for Applications (AKMA) based on 3GPP credentials in the 5G System (5GS).* Technical Specification.

[3] 3GPP. 2024. *System architecture for the 5G System (5GS).* Technical Specification.

[4] Rouzbeh Behnia, Arman Riasi, Mohammadreza Ebrahimi, Sherman SM Chow, Balaji Padmanabhan, and Thang Hoang. 2024. Efficient secure aggregation for privacy-preserving federated machine learning. In *IEEE ACSAC.*

[5] James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. 2020. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security.*

[6] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Computer and Communications Security.*

[7] Dan Boneh, Kevin Lewi, Hart Montgomery, and Ananth Raghunathan. 2013. Key homomorphic PRFs and their applications. In *Annual Cryptology Conference.*

[8] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2019. Abides: Towards high-fidelity market simulation for ai research. *arXiv preprint.*

[9] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19).*

[10] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21).*

[11] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. 2021. Ascon v1. 2: Lightweight authenticated encryption and hashing. *Journal of Cryptology.*

[12] Yue Guo, Antigoni Polychroniadou, Elaine Shi, David Byrd, and Tucker Balch. 2024. MicroSecAgg: Streamlined Single-Server Secure Aggregation. *Proceedings on Privacy Enhancing Technologies.*

[13] Abdul Jabbar, Justin P Rohrer, Victor S Frost, and James PG Sterbenz. 2011. Survivable millimeter-wave mesh networks. *Computer Communications.*

[14] Harish Karthikeyan and Antigoni Polychroniadou. 2024. OPA: one-shot private aggregation with single client interaction and its applications to federated learning. In *International Workshop on Federated Foundation Models in Conjunction with NeurIPS 2024.*

[15] Lianjun Li, Hao Chen, Hao-Hsuan Chang, and Lingjia Liu. 2019. Deep residual learning meets OFDM channel estimation. *IEEE Wireless Communications Letters.*

[16] Marc Lichtman, Raghunandan Rao, Vuk Marojevic, Jeffrey Reed, and Roger Piqueras Jover. 2018. 5G NR jamming, spoofing, and sniffing: Threat assessment and mitigation. In *2018 IEEE international conference on communications workshops.*

[17] Yi Liu, Jialiang Peng, Jiawen Kang, Abdullah M Iliyasu, Dusit Niyato, and Ahmed A Abd El-Latif. 2020. A secure federated learning framework for 5G networks. *IEEE Wireless Communications.*

[18] Yiping Ma, Jess Woods, Sebastian Angel, Antigoni Polychroniadou, and Tal Rabin. 2023. Flamingo: Multi-round single-server secure aggregation with applications to private federated learning. In *IEEE Symposium on Security and Privacy.*

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics.*

[20] Vincent Omollo Nyangaresi, Zaid Ameen Abduljabbar, Mustafa A Al Sibahee, Ayad Ibrahim, Ali Noah Yahya, Iman Qays Abduljaleel, and Enas Wahab Abood. 2022. Optimized hysteresis region authenticated handover for 5G HetNets. In *Artificial Intelligence and Sustainable Computing: Proceedings of ICSISCET 2021.*

[21] Vasileios Perifanis, Nikolaos Pavlidis, Remous-Aris Koutsiamanis, and Pavlos S Efraimidis. 2023. Federated learning for 5G base station traffic forecasting. *Computer Networks.*

[22] Adi Shamir. 1979. How to share a secret. *Commun. ACM.*

[23] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP).*

[24] Nguyen H Tran, Wei Bao, Albert Zomaya, Minh NH Nguyen, and Choong Seon Hong. 2019. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019.*