

Post-Quantum Dropout-Resilient Verifiable Secure Aggregation for Federated Learning

Fateme Sadat Azimi, Hossein Pilaram, and Javad Mohajeri

Abstract—Secure weighted aggregation is essential for privacy-preserving federated learning, yet existing schemes do not simultaneously provide post-quantum security, resilience to client dropouts, verifiability against malicious servers, and resistance to bounded collusion between the server and clients. This paper presents PQ-DVSA, a verifiable secure aggregation scheme that provides post-quantum security and resilience to client dropouts. PQ-DVSA protects both clients’ updated local models and aggregation weights, enables reconstruction of the aggregate mask for the active set after client dropouts, and allows clients to verify the server’s aggregation result under bounded collusion. Experimental results show that PQ-DVSA achieves learning performance comparable to exact weighted aggregation and reduces the runtime for encrypted submission at each client by up to 96% compared with SWAS, the closest existing scheme to PQ-DVSA.

Index Terms—Privacy-preserving federated learning, secure weighted aggregation, post-quantum verifiable aggregation, dropout resilience, Ring Learning With Errors (RLWE).

I. INTRODUCTION

Federated learning (FL) enables multiple clients to collaboratively train a model without transmitting their raw data to a central server [1]. Nevertheless, clients must transmit their updated local models for aggregation, which can reveal information about local training data, as demonstrated by gradient inversion attacks [2]. Therefore, protecting updated local models during aggregation is essential for privacy-preserving FL. Secure aggregation protects individual client contributions while allowing the server to obtain their aggregate. The Secure Aggregation (SecAgg) protocol [3], one of the earliest practical approaches in this direction, combines double masking with Shamir secret sharing [4] to protect individual client updates while providing robustness against client dropouts. However, when the server is malicious, protecting client information alone is insufficient; clients must also be able to verify the aggregation result. VFL [5] introduced verification based on Lagrange interpolation, while VCD-FL [6] further combines grouping and commitment mechanisms to provide dynamic and collusion-resistant verification.

Weighted aggregation introduces an additional privacy concern because FedAvg [1] weights are commonly derived from local data sizes, which may reveal data volume information [7]. SWAS [7], the closest scheme to our work, provides verifiable weighted aggregation while protecting updated local models and local data sizes. However, SWAS [7] does not support

client dropouts or post-quantum security and does not provide verification under bounded collusion.

Protecting FL in the long term also requires security against quantum adversaries. Shor’s algorithm threatens cryptographic constructions based on integer factorization and discrete logarithms [8], motivating the use of lattice-based assumptions such as Learning With Errors (LWE) [9] and Ring Learning With Errors (RLWE) [10]. SLAP [11] provides private aggregation based on RLWE, while LHAgg [12] and PQSF [13] support post-quantum aggregation with resilience to client dropouts. However, these schemes do not jointly provide weighted aggregation, verifiability against malicious servers, and resistance to bounded collusion between the server and clients.

Our proposed Post-Quantum Dropout-Resilient Verifiable Secure Aggregation (PQ-DVSA) scheme inherits the weighted verification approach of SWAS [7] and redesigns its masking and recovery mechanisms to provide post-quantum security, resilience to client dropouts, and verification under bounded collusion. It uses bounded bucketed weights, reconstructs only the aggregate masks of active clients, and independently protects publicly posted polynomial coefficients. Table I summarizes the main features of closely related schemes and compares them with our proposed PQ-DVSA.

The main contributions of this work are as follows.

- We present PQ-DVSA, a verifiable secure weighted aggregation scheme that provides post-quantum security, resilience to client dropouts, and verification under bounded collusion between the server and clients.
- We replace exact data size weights with bounded bucketed weights, reducing the plaintext size required to protect updated local models and aggregation weights while limiting the influence of unusually large reported data sizes.
- We separate the masking of aggregation data from the masking of publicly posted polynomial coefficients, enabling verifiable coefficient aggregation while preserving the confidentiality of protected values.
- We adapt a lattice based multi-stage secret sharing scheme [14] to reconstruct only the aggregate masks of active clients, enabling mask removal after client dropouts without reconstructing individual masking secrets.

The rest of this paper is organized as follows: Sections II–IV review related work, preliminaries, and the SWAS protocol [7], respectively. Sections V–VII present the system model, building blocks, and our PQ-DVSA construction, respectively. Section VIII presents the security analysis, Section IX reports the evaluation, and Section X concludes the paper.

Fateme Sadat Azimi is with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: fs.azimii@ee.sharif.edu). Hossein Pilaram and Javad Mohajeri are with the Electronics Research Institute, Sharif University of Technology, Tehran, Iran (e-mails: pilaram@sharif.edu and mohajer@sharif.edu, respectively).

TABLE I
HIGH-LEVEL COMPARISON OF CLOSELY RELATED SECURE AGGREGATION SCHEMES.

Scheme	Masking Mechanism	Post-Quantum Security	Dropout Res.	Secure Weighted Agg.	Verifiable	Collusion-Res. Verif.
SLAP [11]	RLWE-based PSA encryption	✓	✗	✗	✗	✗
VCD-FL [6]	Pairwise PRG masking	✗	✓	✗	✓	✓
LHAgg [12]	RLWE-HPRG masking	✓	✓	✗	✗	✗
PQSF [13]	Pairwise and self-masking	✓	✓	✗	✗	✗
SWAS [7]	Pairwise PRG masking	✗	✗	Exact	✓	✗
Proposed PQ-DVSA	RLWE-HPRG masking	✓	✓	Bucketed	✓	✓

II. RELATED WORK

This section reviews the work most closely related to ours along two directions: verifiable secure aggregation and post-quantum dropout-resilient aggregation.

A. Verifiable Secure Aggregation

Secure aggregation first focused on hiding individual client updates during aggregation. SecAgg [3] provides this protection through masking and secret sharing while also tolerating client dropouts. As privacy alone does not ensure that the server performs aggregation correctly, subsequent work incorporated explicit verification. VerifyNet [15] uses homomorphic hashing to verify aggregated results, and VeriFL [16] reduces the associated communication cost through linear homomorphic hashing and commitments. Another approach is taken by Zhang et al. [17], who combine privacy protection with Paillier-based verification.

Verification based on polynomial interpolation forms another closely related direction. VFL [5] embeds verification information into Lagrange interpolation, while VCD-FL [6] extends this approach to dynamic participation and collusion-resistant verification. Alongside these developments, VOSA [18] studies verifiable and oblivious aggregation, and MRNIVA [19] addresses non-interactive verification against malicious behavior. Recent work continues to strengthen this line, with Yao et al. [20] considering verifiable secure aggregation in FL networks and VeCoSA-FI [21] incorporating collusion resistance together with incentives. In the weighted setting, SWAS [7] protects both weighted model information and local data-size information while supporting result verification.

B. Post-Quantum Aggregation and Dropout Resilience

A parallel line of research considers aggregation mechanisms that remain secure against quantum-capable adversaries. LaPS [22] is an early RLWE-based construction for private aggregation, while Yang et al. [23] investigate post-quantum secure aggregation specifically for FL. At the same time, improving dropout recovery has motivated more efficient masking mechanisms. HomoAgg [24] uses homomorphic pseudorandom masking and aggregate-secret reconstruction to reduce recovery overhead, although its underlying cryptographic assumptions remain classical.

Lattice-based constructions provide a natural way to combine efficient aggregation with post-quantum security. SLAP [11]

TABLE II
MAIN NOTATION USED IN PQ-DVSA

Notation	Description
λ	Security parameter.
$\mathcal{U} = \{u_1, \dots, u_N\}$	Registered users.
$\mathcal{U}_t, \mathcal{L}_t$	Selected and active users in round t .
$\mathcal{C}$	Corrupted users colluding with the server.
$T, \mathcal{R}_t$	Reconstruction threshold and reconstruction set, with $ \mathcal{R}_t \geq T$.
n'	Upper bound on selected or active users in one round.
d	Model dimension.
M, G	Group size and number of groups, $G = \lceil d/M \rceil$.
τ, p, q	Plaintext, HPRG-mask, and polynomial-mask moduli.
Δ	Scaling factor, $\Delta = \lceil p/\tau \rceil$.
$B, w_{\min}, w_{\max}$	Bucket size and clipping bounds.
w_i	Bounded bucketed weight of user u_i .
b_i, n_i	p domain and q domain masking secrets of user u_i .
N_R	RLWE ring dimension.
$\mathbf{x}_{i,t}, \mathbf{y}_{i,t}$	Quantized update and bucketed weighted update in round t .
$\hat{w}_t, \hat{\mathbf{y}}_t, \hat{\mathbf{x}}_t$	Aggregate weight, aggregate weighted update, and final averaged update.

builds private aggregation directly from RLWE, avoiding the complexity of treating general homomorphic encryption as a black-box component. Extending this direction to dropout-resilient FL, LHAgg [12] combines an RLWE-based HPRG with secret sharing for mask recovery. PQSF [13] instead couples double masking with lattice-based multi-stage secret sharing [14] to support post-quantum aggregation under client dropouts. More recent efforts continue to improve practicality: Standing Firm [25] targets single-round dropout-resilient aggregation, Beskar [26] considers post-quantum protection across a broader private FL pipeline, and Qin et al. [27] study efficient post-quantum aggregation in cross-silo FL.

Overall, the first line of work has progressively strengthened verifiability, collusion resistance, and secure weighting, whereas the second has focused on post-quantum security and resilience to client dropouts. These properties, however, have not been jointly provided with secure weighted aggregation and verification under bounded server–client collusion.

III. PRELIMINARIES AND NOTATION

For readability, Table II lists the main recurring notation; local protocol variables are defined where they appear.

A. Algebraic Setting

Our proposed PQ-DVSA scheme uses three modular domains. Plaintext values are represented modulo τ . Masked weighted

updates, bucketed weights, and commitments are handled modulo p . Polynomial coefficients are defined modulo p and lifted to $\mathbb{Z}_q$ for polynomial masking and aggregation. Following the parameter bounds in [11], the prime moduli p and q satisfy

$$3n'\tau^2 < p, \quad 3n'p < q. \quad (1)$$

For any modulus m , $\mathbb{Z}_m$ denotes the ring of integers modulo m , and $[a]_m$ denotes reduction modulo m . The lattice-based masking layers use the polynomial rings

$$R_p = \mathbb{Z}_p[X]/(X^{N_R} + 1), \quad R_q = \mathbb{Z}_q[X]/(X^{N_R} + 1), \quad (2)$$

where N_R is the ring dimension.

B. RLWE-Based HPRG

We use an RLWE-based almost-homomorphic pseudorandom generator (HPRG) to generate aggregation masks [12]. For secrets s_1 and s_2 , its outputs satisfy

$$\text{HPRG}(s_1 + s_2) = \text{HPRG}(s_1) + \text{HPRG}(s_2) + \mathbf{e}, \quad (3)$$

where $\mathbf{e}$ is the residual error arising from the underlying RLWE samples. In our instantiation, individual error coefficients are sampled from a centered multinomial distribution. The HPRG's approximate additivity allows users to remove the aggregate mask using the sum of the masking secrets; correct decoding requires the accumulated error to satisfy (79). Our treatment of message scaling and noise follows SLAP [11].

The pseudorandomness of the masks relies on the decisional RLWE assumption. For a uniformly sampled a , a secret s , and an error polynomial e drawn from a suitable small-error distribution, samples of the form $(a, [a \cdot s + e]_p)$ are computationally indistinguishable from uniform for appropriately chosen parameters [10].

C. Multi-Stage Secret Sharing

Multi-stage secret sharing (MSSS) allows multiple secrets to be reconstructed at different stages while unreconstructed secrets remain computationally protected [14]. In a (T, N) -threshold setting, reconstruction requires at least T participants. Its multi-use property allows the public reconstruction information to be renewed for new secrets without redistributing the private shares [13], [14].

For modulus η , threshold T , users $\mathcal{U}$, and secrets $\{s_j\}_{j=1}^J$, the dealer generates secret-specific public information PP_j and a reusable private share SH_i for each user:

$$(\{PP_j\}_{j=1}^J, \{SH_i\}_{i=1}^N) \leftarrow \text{MSSS-Share}(\eta, T, \mathcal{U}, \{s_j\}_{j=1}^J). \quad (4)$$

To reconstruct s_j , each user u_i in a reconstruction set $\mathcal{R}$ derives a pseudo-share

$$\text{msg}_{i,j} \leftarrow \text{MSSS-Msg}(SH_i, PP_j), \quad u_i \in \mathcal{R}. \quad (5)$$

A pseudo-share is a reconstruction message derived from the user's private share and the public information of the target secret; it enables reconstruction without revealing the reusable private share. Given $|\mathcal{R}| \geq T$, the secret is recovered as

$$s_j \leftarrow \text{MSSS-Recon}(PP_j, \mathcal{R}, \{\text{msg}_{i,j}\}_{u_i \in \mathcal{R}}). \quad (6)$$

For a refreshed secret s'_j , only its public reconstruction information is renewed:

$$PP'_j \leftarrow \text{MSSS-Renew}(PP_j, s'_j), \quad (7)$$

while the private shares remain unchanged. Section VI-B describes how this interface is adapted for aggregate-mask reconstruction in PQ-DVSA.

D. Lagrange Interpolation

For distinct points $z_1, \dots, z_{M+1} \in \mathbb{F}_p$ and values $v_1, \dots, v_{M+1} \in \mathbb{F}_p$, the unique polynomial $f \in \mathbb{F}_p[X]$ of degree at most M satisfying $f(z_j) = v_j$ is

$$f(X) = \sum_{j=1}^{M+1} v_j \ell_j(X), \quad \ell_j(X) = \prod_{\substack{h=1 \\ h \neq j}}^{M+1} \frac{X - z_h}{z_j - z_h}. \quad (8)$$

In PQ-DVSA, each polynomial encodes M masked model components and one auxiliary verification value.

IV. THE SWAS PROTOCOL

This section reviews SWAS [7], the baseline of PQ-DVSA. SWAS provides secure and verifiable weighted aggregation through pairwise masking, Lagrange interpolation, CRT-based recovery of the aggregate data size, and HPRG-based verification. It assumes a trusted authority (TA), honest-but-curious users, and a malicious server.

A. Initialization

The TA selects a grouping parameter M and partitions a d -dimensional model into G groups. Then, he pads the last group with zeros if needed.

$$G = \left\lceil \frac{d}{M} \right\rceil \quad (9)$$

PQ-DVSA later uses the same grouping principle to organize model coordinates into polynomial blocks.

For the N registered clients, the TA generates signing key pairs and a symmetric seed matrix $S = (s_{ij})_{i,j=1}^N$, with $s_{ij} = s_{ji}$ and unused diagonal entries. The parameters M , the pairwise coprime CRT moduli $\{g_j\}_{j=1}^{G-1}$, the shared seed s_z , PRG, and HPRG are common to all clients, while each client u_i securely receives only its signing secret key sk_i and seed row $S[i]$. After delivering the initial global model to the first-round clients, the TA remains inactive during normal training.

B. Local Computation and Encoding

In round t , client u_i obtains a local update x_i^t and computes the exact weighted update

$$y_i = \text{num}_i x_i^t, \quad \text{num}_i = |\mathcal{D}_i|, \quad (10)$$

where num_i is the local data size. The weighted update is masked with pairwise PRG values:

$$y'_i = y_i + \sum_{u_j \in \mathcal{U}_t, i < j} \text{PRG}(s_{ij}) - \sum_{u_j \in \mathcal{U}_t, i > j} \text{PRG}(s_{ij}). \quad (11)$$

The local data size is also masked by a zero-sum share:

$$\text{num}'_i = \text{num}_i + [[0]]_i, \quad \sum_{u_i \in \mathcal{U}_t} [[0]]_i = 0. \quad (12)$$

For verification, SWAS encodes num'_i as CRT residues and appends an HPRG value:

$$a_j^{(i)} \equiv \text{num}'_i \pmod{g_j}, \quad (13)$$

$$A_i = (a_1^{(i)}, \dots, a_{G-1}^{(i)}, \text{HPRG}(\text{num}_i)).$$

Using points derived from $\text{PRG}(s_z \parallel t)$, each client builds one degree- M Lagrange polynomial per group, containing M masked model values and one verification entry from A_i . The uploaded coefficient vector is

$$C_i = (C_{i,1}, \dots, C_{i,G}). \quad (14)$$

C. Cloud Aggregation

The CS aggregates the uploaded coefficients group-wise:

$$C_g^* = \sum_{u_i \in \mathcal{U}_t} C_{i,g}, \quad g = 1, \dots, G. \quad (15)$$

D. Client Verification and Model Reconstruction

Clients evaluate the aggregate polynomials to recover the aggregate weighted update $\hat{y}$ and the aggregate verification vector. From the CRT residues $\hat{A}_j$, they recover the aggregate data size as

$$\text{num}^t = \sum_{j=1}^{G-1} \hat{A}_j \hat{c}_j \Gamma_j \pmod{\Gamma}, \quad (16)$$

where

$$\Gamma = \prod_{j=1}^{G-1} g_j, \quad \Gamma_j = \Gamma/g_j, \quad \hat{c}_j = \Gamma_j^{-1} \pmod{g_j}. \quad (17)$$

If the final verification entry matches $\text{HPRG}(\text{num}^t)$, clients recover the aggregated model update as

$$x^t = \frac{1}{\text{num}^t} \hat{y} \quad (18)$$

and sign the accepted aggregate.

E. Practical Challenges and Limitations of SWAS

SWAS is secure under its stated threat model [7], but its verification mechanism is not designed for the stronger PQ-DVSA setting, where interpolation points may become known through bounded server-client collusion. In SWAS, these points are derived from the shared seed s_z . If a corrupted client discloses s_z , or the corresponding round points, then for each group g the server obtains

$$Z_{g,t} = (z_{g,1}^{(t)}, \dots, z_{g,M+1}^{(t)}) \leftarrow \text{PRG}(s_z \parallel t). \quad (19)$$

The server can then evaluate the uploaded polynomials at these points and observe the encoded masked model values and the auxiliary verification entry. Although this does not remove the pairwise masks, it reveals the interpolation structure used for verification.

Let $F_g^{(t)}(X)$ denote the aggregate polynomial represented by C_g^* . Once $Z_{g,t}$ is known, a malicious server could return

$$\begin{aligned} \tilde{F}_g^{(t)}(X) &= F_g^{(t)}(X) + \Delta_g(X), \\ \Delta_g(z_{g,M+1}^{(t)}) &= 0, \quad \Delta_g(z_{g,\ell}^{(t)}) \neq 0, \quad 1 \leq \ell \leq M. \end{aligned} \quad (20)$$

In this case, the auxiliary verification value is preserved while the model-coordinate evaluations are modified. Thus, disclosure of the interpolation points can weaken the original verification mechanism; in particular, SWAS does not by itself provide collusion-resistant verification in this stronger setting.

The same exposure may also lead to cross-round leakage. In (11), the pairwise masks depend on the fixed seeds s_{ij} and the selected set, but not on the round index. Hence, if the same selected set participates in two rounds and the interpolation points are known, the server can obtain $y_i^{(t)}$ and $y_i^{(t-1)}$, whose difference cancels the repeated pairwise masks:

$$y_i^{(t)} - y_i^{(t-1)} = y_i^{(t)} - y_i^{(t-1)}. \quad (21)$$

This reveals the difference between the user's weighted updates across the two rounds. PQ-DVSA instead uses round-separated p -domain masks for encoded values and round-separated q -domain masks for posted constant coefficients, as specified in (27) and (55).

Exact data-size weights can provide another inference channel if aggregate data sizes become available to colluding clients:

$$\text{num}^t = \sum_{u_i \in \mathcal{U}_t} \text{num}_i. \quad (22)$$

Across rounds, these observations can be written as $\mathbf{N} = \mathbf{P}\mathbf{n}$, where $\mathbf{n} = (\text{num}_1, \dots, \text{num}_N)$ and $P_{t,i} = 1$ if $u_i \in \mathcal{U}_t$. With sufficiently many independent participation patterns, individual data sizes may be recovered or substantially narrowed. These observations motivate bounded bucketed weights, non-repeated selected/active-set pairs, q -domain masking of posted coefficients, and active-set aggregate-mask reconstruction. Their construction is given in Section VII, and their security is analyzed in Section VIII.

V. SYSTEM MODEL AND OVERVIEW

Our proposed PQ-DVSA scheme supports secure bucketed weighted aggregation in FL with a malicious server, user dropouts, and bounded server-client collusion. It involves a trusted authority (TA), a server, and N users.

A. Entities and Threat Model

The TA is trusted and participates only in initialization. It generates the public parameters, masking secrets, and private MSSS shares, announces the first selected set $\mathcal{U}_1$, and then remains inactive.

The server is malicious and performs aggregation. It may omit, modify, or selectively aggregate submissions, return stale or incorrect results, exploit dropouts, or collude with a set $\mathcal{C}$ of protocol compliant users, where $|\mathcal{C}| < T$. Corrupted users reveal their local states and secrets to the server but do not generate malformed protocol messages.

For $t \geq 2$, the server proposes $\mathcal{U}_t$, while actual participation also depends on the local eligibility check and user availability. After submission, the server announces the active set $\mathcal{L}_t \subseteq \mathcal{U}_t$. Each user independently recomputes $\mathcal{L}_t$ from the authenticated bulletin board and rejects any inconsistent announcement. The bulletin board authenticates the origin of each posting and provides an append only, globally consistent view. Reconstruction uses $\mathcal{R}_t \subseteq \mathcal{L}_t$ with $|\mathcal{R}_t| \geq T$.

Under this model, PQ-DVSA protects honest users' updates, bucketed weights, verification data, and individual masking secrets, while preventing acceptance of an aggregate inconsistent with the authenticated postings and declared active set.

B. Design Rationale

PQ-DVSA addresses three limitations of SWAS in stronger deployment settings: client dropouts, public interpolation, and bounded server-client collusion. It retains the weighted-verification framework while modifying the treatment of weights, masks, and coefficient integrity.

First, exact data-size weights are replaced by bounded bucketed weights. User u_i computes

$$\tilde{w}_i = \left\lceil \frac{\text{num}_i}{B} \right\rceil, \quad (23)$$

and clips it as

$$w_i = \begin{cases} w_{\min}, & \tilde{w}_i < w_{\min}, \\ \tilde{w}_i, & w_{\min} \leq \tilde{w}_i \leq w_{\max}, \\ w_{\max}, & \tilde{w}_i > w_{\max}. \end{cases} \quad (24)$$

Thus $w_{\min} \leq w_i \leq w_{\max}$, limiting the protected plaintext range and the impact of unusually large data-size claims.

Second, qualified users reconstruct only the aggregate masking material of the declared active set. This suffices for demasking while keeping each honest user's individual masking secrets hidden from a coalition with fewer than T private shares.

Third, PQ-DVSA separates masking domains:

$$b_i \in \mathbb{Z}_p^T, \quad n_i \in \mathbb{Z}_q^T, \quad q \gg p \gg \tau. \quad (25)$$

The vector b_i derives p -domain HPRG masks for updates, weights, and commitments; n_i derives q -domain polynomial masks for coefficient posting.

Finally, masked coefficient vectors are posted on an authenticated bulletin board. Hence the server is bound to a publicly checkable coefficient sum, while privacy still relies on the two-domain masks rather than on hidden interpolation points.

VI. BUILDING BLOCKS USED IN PQ-DVSA

A. RLWE-Based HPRG Instantiation

PQ-DVSA uses the RLWE-based almost-homomorphic PRG of [12], adapted to the SLAP message-scaling framework [11]. The factor $\Delta = \lceil p/\tau \rceil$ embeds plaintexts in $\mathbb{Z}_p$, and centered multinomial noise supports efficient sampling. To align the MSSS secret dimension with the HPRG input structure, we choose the HPRG block length m such that $m \mid T$, and set $r = T/m$. Each p -domain MSSS secret is then written as

$$b_i = (b_i^{(1)}, \dots, b_i^{(r)}), \quad b_i^{(s)} \in R_p^m. \quad (26)$$

The first $r-2$ chunks mask model coordinates, while $b_i^{(r-1)}$ and $b_i^{(r)}$ mask the bucketed weight and commitment. For coordinate j ,

$$\text{HPRG}_t^{(b)}(b_i)[j] = \left[\left\langle a_{j,t}^{(b)}, b_i^{(p_b(j))} \right\rangle + e_{i,j,t}^{(b)} \right]_p, \quad (27)$$

where $a_{j,t}^{(b)}$ is domain-separated by round and coordinate. The masked coordinate and weight are

$$y'_{i,t}[j] = \left[\Delta y_{i,t}[j] + \text{HPRG}_t^{(b)}(b_i)[j] \right]_p, \quad (28)$$

$$w'_i = \left[\Delta w_i + \text{HPRG}_t^{(w)}(b_i) \right]_p. \quad (29)$$

Linearity lets users subtract the public inner product evaluated on $\sum_{u_i \in \mathcal{L}_t} b_i$; only accumulated noise remains before centered lifting and rounding. The same mechanism masks weights and commitments.

B. Adapted MSSS for Dropout Resilience

Building on the interface in Section III-C, our scheme adapts the improved lattice-based MSSS construction used in PQSF [13], which builds on [14]. The adaptation reconstructs only the sum of the masking secrets associated with the declared active set, rather than individual secrets.

a) Sharing phase.: The TA acts as the dealer and generates, for each user u_i , two independent masking secrets $b_i \in \mathbb{Z}_p^T$ and $n_i \in \mathbb{Z}_q^T$. These define two MSSS instances $\xi \in \{b, n\}$, where $\eta_b = p$, $\eta_n = q$, $s_i^{(b)} = b_i$, and $s_i^{(n)} = n_i$. Here, i indexes masking secrets and a indexes share holders.

For each instance ξ , the TA selects $v_\xi \in \mathbb{Z}_{\eta_\xi}^T$ with last entry 1, and for each secret $s_i^{(\xi)}$ constructs $B_{(\xi,i)} \in \mathbb{Z}_{\eta_\xi}^{T \times T}$ such that

$$s_i^{(\xi)} = B_{(\xi,i)} v_\xi \pmod{\eta_\xi}. \quad (30)$$

Let $L_\xi \geq \max\{T \log T, N\}$. The TA selects $\Lambda_\xi = (\lambda_{\xi,1}, \dots, \lambda_{\xi,N}) \in \mathbb{Z}_{\eta_\xi}^{T \times N}$, such that any T columns are linearly independent, and generates reusable private shares $SH_{\xi,a} = (c_{\xi,a}, e_{\xi,a})$, where $c_{\xi,a} \in \mathbb{Z}_{\eta_\xi}^{L_\xi}$ and $e_{\xi,a} \in \mathbb{Z}_{\eta_\xi}^T$.

For each $s_i^{(\xi)}$, the TA constructs $A_{(\xi,i)} \in \mathbb{Z}_{\eta_\xi}^{T \times L_\xi}$ satisfying

$$A_{(\xi,i)} c_{\xi,a} + \mathcal{H}_\xi(e_{\xi,a}) = B_{(\xi,i)} \lambda_{\xi,a} \pmod{\eta_\xi}, \quad (31)$$

for every share holder u_a . The corresponding public information is $PP_{\xi,i} = (A_{(\xi,i)}, v_\xi, \Lambda_\xi, \mathcal{H}_\xi)$. The TA publishes $\{PP_{\xi,i}\}_{i=1}^N$ and privately sends $SH_{\xi,a}$ to u_a .

b) Aggregate reconstruction.: In round t , the reconstruction target for instance ξ is

$$S_{\mathcal{L}_t}^{(\xi)} = \sum_{u_i \in \mathcal{L}_t} s_i^{(\xi)} \pmod{\eta_\xi}. \quad (32)$$

For a reconstruction set $\mathcal{R}_t \subseteq \mathcal{L}_t$ with $|\mathcal{R}_t| \geq T$, each $u_a \in \mathcal{R}_t$ computes the aggregate pseudo-share

$$\begin{aligned} \text{msg}_{\xi,a}^{(t)} &= \sum_{u_i \in \mathcal{L}_t} \text{MSSS-Msg}(SH_{\xi,a}, PP_{\xi,i}) \\ &= \sum_{u_i \in \mathcal{L}_t} (A_{(\xi,i)} c_{\xi,a} + \mathcal{H}_\xi(e_{\xi,a})) \\ &= B_{\mathcal{L}_t}^{(\xi)} \lambda_{\xi,a} \pmod{\eta_\xi}, \end{aligned} \quad (33)$$

where $B_{\mathcal{L}_t}^{(\xi)} = \sum_{u_i \in \mathcal{L}_t} B_{(\xi,i)} \pmod{\eta_\xi}$.

From any T valid pseudo-shares, let $\mathcal{R}_t^* = \{u_{a_1}, \dots, u_{a_T}\}$, $D_{\xi,t} = (\text{msg}_{\xi,a_1}^{(t)}, \dots, \text{msg}_{\xi,a_T}^{(t)})$, and $W_{\xi,t} = (\lambda_{\xi,a_1}, \dots, \lambda_{\xi,a_T})$. Since $W_{\xi,t}$ is invertible, users recover

$$B_{\mathcal{L}_t}^{(\xi)} = D_{\xi,t} W_{\xi,t}^{-1} \pmod{\eta_\xi}, \quad (34)$$

$$S_{\mathcal{L}_t}^{(\xi)} = B_{\mathcal{L}_t}^{(\xi)} v_\xi \pmod{\eta_\xi}. \quad (35)$$

Thus, the two instances recover only $S_{\mathcal{L}_t}^{(b)}$ and $S_{\mathcal{L}_t}^{(n)}$, which suffice for aggregate-mask removal without reconstructing any individual b_i or n_i .

c) *Renewal phase.*: When new masking secrets are provisioned, the corresponding public information is renewed as

$$\{PP'_{\xi,i}\}_{i=1}^N \leftarrow \text{MSSS-Renew} \left(\{PP_{\xi,i}\}_{i=1}^N, \{s_i^{(\xi)'}\}_{i=1}^N \right), \quad (36)$$

while the private shares $SH_{\xi,a}$ remain unchanged. Renewal is therefore not required during ordinary training rounds.

VII. PQ-DVSA PROTOCOL CONSTRUCTION

Our proposed PQ-DVSA scheme consists of four stages: initialization, user computation, server aggregation, and local verification with model reconstruction. The TA announces $\mathcal{U}_1$. For $t \geq 2$, the server proposes a candidate set $\mathcal{U}_t$, while actual participation also depends on the local eligibility check. After submission, the server announces the active set $\mathcal{L}_t \subseteq \mathcal{U}_t$. Users reject any pair $(\mathcal{U}_t, \mathcal{L}_t)$ that appeared in a previously accepted round.

A. Initialization

Initialization has a one-time user/TA setup and a per-round server/user setup before local computation.

One-Time User and TA Setup.

Let $\mathcal{U} = \{u_1, u_2, \dots, u_N\}$. For each u_i , the TA samples $\sigma_{b_i}, \sigma_{n_i} \leftarrow \{0, 1\}^\lambda$ and expands them using domain-separated hash/XOF functions

$$H_b : \{0, 1\}^* \rightarrow \mathbb{Z}_p^T, \quad H_n : \{0, 1\}^* \rightarrow \mathbb{Z}_q^T. \quad (37)$$

The resulting secret vectors are

$$\begin{aligned} b_i &= H_b(\text{tag}_b \parallel \sigma_{b_i}) \in \mathbb{Z}_p^T, \\ n_i &= H_n(\text{tag}_n \parallel \sigma_{n_i}) \in \mathbb{Z}_q^T. \end{aligned} \quad (38)$$

Let PP_b and PP_n denote the public reconstruction information for the p and q MSSS instances, respectively. The TA runs

$$\begin{aligned} (PP_b, \{SH_{b,i}\}_{i=1}^N) &\leftarrow \text{MSSS-Share}(p, T, \mathcal{U}, \{b_i\}_{i=1}^N), \\ (PP_n, \{SH_{n,i}\}_{i=1}^N) &\leftarrow \text{MSSS-Share}(q, T, \mathcal{U}, \{n_i\}_{i=1}^N). \end{aligned} \quad (39)$$

Each user u_i privately receives its MSSS share pairs

$$SH_{b,i} = (c_{b,i}, e_{b,i}), \quad SH_{n,i} = (c_{n,i}, e_{n,i}), \quad (40)$$

together with the seeds $\sigma_{b_i}, \sigma_{n_i}$.

The TA publishes the MSSS public parameters

$$PP_b = (\{A_{(b,i)}\}_{i=1}^N, v_b, \Lambda_b, \mathcal{H}_b), \quad (41)$$

and

$$PP_n = (\{A_{(n,i)}\}_{i=1}^N, v_n, \Lambda_n, \mathcal{H}_n). \quad (42)$$

The TA also publishes the moduli satisfying (1), Δ , B , $w_{\min}$, $w_{\max}$ with $n'w_{\max} < \tau$, M and G as in (9), the public interpolation seed s , the reserved group R , the vector $h \in \{0, 1\}^{G-1}$, and the public masking and commitment parameters.

After this one-time setup, the TA becomes inactive and does not participate in normal training rounds.

Per-Round Server and User Setup. For each round $t \geq 2$, the server announces a candidate selected user set

$$\mathcal{U}_t \subseteq \mathcal{U}, \quad |\mathcal{U}_t| \leq n'. \quad (43)$$

Selected users receive the current global model and participate only if they pass the local parity-based eligibility check and choose to continue.

B. User Computation

Each participating u_i computes its bounded bucketed weight w_i using (24).

a) *Masking Model Parameters.*: Let $\mathbf{x}_{i,t} \in \mathbb{Z}_\tau^d$ be the quantized local update. The bucketed weighted update is

$$\mathbf{y}_{i,t} = [w_i \mathbf{x}_{i,t}]_\tau \in \mathbb{Z}_\tau^d. \quad (44)$$

The user masks $\mathbf{y}_{i,t}$ in $\mathbb{Z}_p$ using b_i as in (28) to obtain $\mathbf{y}'_{i,t}$.

b) *Masking Bucketed Weight and Verification Value.*: The user masks w_i using the reserved chunk $b_i^{(r-1)}$ as in (29) to obtain w'_i . To build $D_i \in \mathbb{Z}_p^G$, the masked value w'_i is split into k shares

$$d_{i,1}, d_{i,2}, \dots, d_{i,k} \in \mathbb{Z}_p, \quad \sum_{\ell=1}^k d_{i,\ell} = w'_i \pmod{p}, \quad (45)$$

where k is the Hamming weight of h . Let $(\rho_1, \dots, \rho_{G-1})$ denote the ordered indices in $\{1, \dots, G\} \setminus \{R\}$. Scanning h , the user places the next share in $D_i[\rho_\ell]$ when $h[\ell] = 1$ and otherwise places the next filler value:

$$D_i[\rho_\ell] = \begin{cases} d_{i,\alpha_\ell}, & h[\ell] = 1, \\ \omega_{i,\beta_\ell}, & h[\ell] = 0, \end{cases} \quad \ell = 1, \dots, G-1. \quad (46)$$

The filler values are chosen such that $\sum_{\ell: h[\ell]=0} \omega_{i,\beta_\ell} = 0 \pmod{p}$. Therefore,

$$\sum_{g \neq R} D_i[g] = w'_i \pmod{p}. \quad (47)$$

At reserved group R , the user inserts a commitment on w_i using chunk $b_i^{(r)}$ and the RLWE-style form [28]. The public parameter is

$$a_t^{(com)} = H_{pp}(\text{tag}_{com} \parallel a_0 \parallel \text{Enc}(t)), \quad (48)$$

and the commitment is

$$c_i = \left[\left\langle a_t^{(com)}, b_i^{(r)} \right\rangle + w_i + \tau e_i^{(com)} \right]_p, \quad (49)$$

where $e_i^{(com)} \leftarrow \chi_{mn}$. The verification vector is completed by setting $D_i[R] = c_i$. Although c_i is computed in the p -domain, it is verified after aggregation and mask removal by reducing the aggregate relation modulo τ ; it checks aggregate bucketed-weight consistency, not a full p -domain opening.

c) *Encoding Model Components and Verification Values.*

From the public seed s and round t , the user derives the interpolation points

$$(z_1, z_2, \dots, z_{M+G}) \leftarrow \text{PRG}(s \parallel t) \in \mathbb{F}_p^{M+G}. \quad (50)$$

The masked vector $\mathbf{y}'_{i,t}$ is partitioned into G groups of size M , with zero padding if needed. For each group g , the user constructs the unique degree- M polynomial $f_{(i,g)}(X) \in \mathbb{F}_p[X]$ satisfying

$$f_{(i,g)}(z_j) = y'_{i,t}[M(g-1) + j], \quad j = 1, \dots, M, \quad (51)$$

and

$$f_{(i,g)}(z_{M+g}) = D_i[g]. \quad (52)$$

The coefficients are obtained by Lagrange interpolation using (8). An inverse NTT may be used when the evaluation points form a compatible domain; otherwise Lagrange interpolation remains the generic construction.

d) *Polynomial Masking.*

$$f_{(i,g)}(X) = \sum_{\ell=0}^M C_{(i,g,\ell)} X^\ell \pmod{p} \quad (53)$$

be the coefficient form of $f_{(i,g)}(X)$. To protect the encoded values against leakage across rounds, the user masks the constant coefficient using the independent secret n_i and the SLAP masking construction [11].

The vector n_i is interpreted as

$$n_i = (n_i^{(1)}, n_i^{(2)}, \dots, n_i^{(r_n)}), \quad n_i^{(s)} \in R_q^m. \quad (54)$$

Let $\rho_\mu : \{1, \dots, G\} \rightarrow \{1, \dots, r_n\}$ be the public chunk selection map. For each group g ,

$$\mu_{i,g}^{(t)} = \left[\left\langle a_{g,t}^{(\mu)}, n_i^{(\rho_\mu(g))} \right\rangle + pe_{i,g,t}^{(\mu)} \right]_q, \quad (55)$$

where $a_{g,t}^{(\mu)} \in R_q^m$ is domain-separated for polynomial masking and $e_{i,g,t}^{(\mu)}$ is sampled noise. The masked constant is

$$\tilde{C}_{(i,g,0)}^{(t)} = [C_{(i,g,0)} + \mu_{i,g}^{(t)}]_q. \quad (56)$$

For $\ell = 1, \dots, M$, remaining coefficients are embedded into $\mathbb{Z}_q$ using $\iota_{p \rightarrow q} : \mathbb{Z}_p \rightarrow \mathbb{Z}_q$:

$$\bar{C}_{(i,g,\ell)} = \iota_{p \rightarrow q}(C_{(i,g,\ell)}) \in \mathbb{Z}_q. \quad (57)$$

The transmitted coefficient vector for group g is

$$C_{(i,g)} = \left(\tilde{C}_{(i,g,0)}^{(t)}, \bar{C}_{(i,g,1)}, \dots, \bar{C}_{(i,g,M)} \right) \in \mathbb{Z}_q^{M+1}. \quad (58)$$

Finally, u_i forms

$$C_i = (C_{(i,1)}, C_{(i,2)}, \dots, C_{(i,G)}) \quad (59)$$

and publishes (t, u_i, C_i) on the authenticated bulletin board.

C. *Server-Side Aggregation*

In round t , participating users post their masked coefficient vectors. The server forms $\mathcal{L}_t \subseteq \mathcal{U}_t$ from valid postings and announces it in the public transcript. Users reject malformed, duplicated, or inconsistent postings. The protocol proceeds only if $|\mathcal{L}_t| \geq T$; otherwise the round aborts.

All posted coefficients lie in $\mathbb{Z}_q$. For each group g and index ℓ , the server sums bulletin-board coefficients modulo q :

$$C_{(\cdot,g,\ell)}^{(t)} = \left[\sum_{u_i \in \mathcal{L}_t} C_{(i,g,\ell)} \right]_q. \quad (60)$$

The aggregate coefficient vector is

$$C^{(t)} = \left(C_{(\cdot,1)}^{(t)}, C_{(\cdot,2)}^{(t)}, \dots, C_{(\cdot,G)}^{(t)} \right). \quad (61)$$

The server broadcasts $C^{(t)}$. Since all summands are authenticated bulletin board postings, accepting users can independently verify the coefficientwise aggregate before mask removal.

D. *Local Verification and Model Reconstruction*

Each user first verifies that the announced active set $\mathcal{L}_t$ matches the valid postings on the authenticated bulletin board. Each reconstruction user $u_a \in \mathcal{R}_t \subseteq \mathcal{L}_t$ then publishes $(\text{msg}_{b,a}^{(t)}, \text{msg}_{n,a}^{(t)})$.

a) *Coefficient Verification.* Every accepting user verifies

$$C^{(t)} \stackrel{?}{=} \left[\sum_{u_i \in \mathcal{L}_t} C_i \right]_q. \quad (62)$$

The user rejects if the equality fails or if any member of $\mathcal{L}_t$ lacks a valid posting. This check binds the server to the authenticated coefficient vectors of the declared active set.

b) *Aggregate Mask Reconstruction.* Using any T valid pairs of $(\text{msg}_{b,a}^{(t)}, \text{msg}_{n,a}^{(t)})$, users reconstruct

$$S_b^{(t)} = \sum_{u_i \in \mathcal{L}_t} b_i \pmod{p}, \quad S_n^{(t)} = \sum_{u_i \in \mathcal{L}_t} n_i \pmod{q}. \quad (63)$$

Here $S_b^{(t)}$ removes HPRG masks for updates, weights, and commitments, while $S_n^{(t)}$ removes q -domain polynomial masks; $S_b^{(t,s)}$ and $S_n^{(t,s)}$ denote reconstructed chunks.

c) *Polynomial Unmasking.* For each group g , users subtract the aggregate linear mask from the constant coefficient in $\mathbb{Z}_q$, take its centered representative, and then reduce modulo p . Under the prescribed decoding bound, this removes the accumulated p -multiple noise:

$$C_{(\cdot,g,0)}^{p,t} = \left[\left\langle \left[C_{(\cdot,g,0)}^{(t)} - \left\langle a_{g,t}^{(\mu)}, S_n^{(t,\rho_\mu(g))} \right\rangle \right]_q, \right\rangle_q \right]_p. \quad (64)$$

For $x \in \mathbb{Z}_q$, $\langle x \rangle_q$ denotes its centered representative in $(-q/2, q/2]$.

For $\ell = 1, \dots, M$, the remaining coefficients are reduced directly to $\mathbb{Z}_p$ as $C_{(\cdot,g,\ell)}^{p,t} = [C_{(\cdot,g,\ell)}^{(t)}]_p$. The resulting aggregate polynomial is

$$F_g^{(t)}(X) = \sum_{\ell=0}^M C_{(\cdot,g,\ell)}^{p,t} X^\ell \pmod{p}. \quad (65)$$

d) *Bucketed-Weight and Commitment Verification.*: After (62), users verify the aggregate bucketed weight and commitment by evaluating each polynomial at its auxiliary point:

$$\widehat{D}_g^{(t)} = F_g^{(t)}(z_{M+g}) \pmod{p}, \quad g = 1, \dots, G. \quad (66)$$

The reserved group R carries the commitment and is excluded; the remaining values give the aggregate masked bucketed weight:

$$w'_t = \left[\sum_{\substack{g=1 \\ g \neq R}}^G \widehat{D}_g^{(t)} \right]_p, \quad (67)$$

where w'_t is masked. Users remove the HPRG mask and decode by centered lifting and scaling:

$$\widehat{w}_t = \text{Round} \left(\frac{\left\langle \left[w'_t - \left\langle a_t^{(w)}, S_b^{(t,r-1)} \right\rangle \right]_p \right\rangle_p}{\Delta} \right). \quad (68)$$

For $x \in \mathbb{Z}_p$, $\langle x \rangle_p$ is the centered lift to $(-p/2, p/2]$, used before decoding scaled plaintexts.

For commitment verification, users unmask the reserved value $\widehat{D}_R^{(t)}$:

$$z_R^{(t)} = \left\langle \left[\widehat{D}_R^{(t)} - \left\langle a_t^{(com)}, S_b^{(t,r)} \right\rangle \right]_p \right\rangle_p. \quad (69)$$

Since the commitment noise is a multiple of τ , users verify

$$[z_R^{(t)}]_\tau \stackrel{?}{=} \widehat{w}_t. \quad (70)$$

Failure aborts the round; otherwise the aggregate bucketed weight is accepted.

e) *Model Reconstruction.*: After verification, users reconstruct the aggregate bucketed weighted update. For $j = 1, \dots, M$, let $\ell = M(g-1) + j$. If $\ell \leq d$, $\widehat{y}'_t[\ell] = F_g^{(t)}(z_j)$; padded coordinates are discarded. Users then demask and decode

$$\widehat{y}_t[\ell] = \text{Round} \left(\frac{\left\langle \left[\widehat{y}'_t[\ell] - \left\langle a_{\ell,t}^{(b)}, S_b^{(t,\rho_b(\ell))} \right\rangle \right]_p \right\rangle_p}{\Delta} \right). \quad (71)$$

The normalized aggregate update is

$$\widehat{\mathbf{x}}_t = \frac{1}{\widehat{w}_t} \widehat{\mathbf{y}}_t. \quad (72)$$

f) *Next-Round Client Eligibility.*: After round t is accepted, the public transcript on the authenticated bulletin board determines

$$\rho_t = \text{Hash} \left(\text{tag}_{\text{elig}} \parallel t \parallel \mathcal{U}_t \parallel \mathcal{L}_t \parallel C^{(t)} \right). \quad (73)$$

For round $t+1$, the server announces a candidate set $\mathcal{U}_{t+1}$. Each candidate u_j locally computes

$$\beta_{j,t+1} = \text{LSB}(\text{Hash}(\rho_t \parallel n_j)). \quad (74)$$

User u_j participates only if

$$\beta_{j,t+1} = \text{LSB}(\text{ID}_j). \quad (75)$$

Since n_j is private, the server cannot determine in advance which candidates satisfy the eligibility condition. Thus, the server proposes the candidate set, while the local check limits its control over actual participation.

VIII. SECURITY ANALYSIS

We analyze our proposed PQ-DVSA scheme under the malicious-server model described in Subsection V-A. The server may collude with a coalition $\mathcal{C}$ of corrupted users satisfying $|\mathcal{C}| < T$, while each declared active set $\mathcal{L}_t$ satisfies $|\mathcal{L}_t| \geq T$. The analysis relies on the RLWE-based pseudorandomness of the masking mechanisms [10], [12], the threshold privacy of the lattice-based MSSS construction [13], [14], and authenticated, globally consistent bulletin-board postings. We establish correctness, privacy under bounded collusion, and verifiability.

A. Correctness

Theorem 1 (Correctness). Consider an honestly executed round t with active set $\mathcal{L}_t$ and reconstruction set $\mathcal{R}_t \subseteq \mathcal{L}_t$, where $|\mathcal{R}_t| \geq T$. Under the parameter bounds in (1) and the decoding conditions specified below, every accepting user recovers

$$\widehat{\mathbf{x}}_t = \frac{\sum_{u_i \in \mathcal{L}_t} w_i \mathbf{x}_{i,t}}{\sum_{u_i \in \mathcal{L}_t} w_i}. \quad (76)$$

Proof. By correctness of the adapted MSSS construction, any T valid reconstruction messages recover the active-set aggregate masking secrets $S_b^{(t)} = \sum_{u_i \in \mathcal{L}_t} b_i \pmod{p}$ and $S_n^{(t)} = \sum_{u_i \in \mathcal{L}_t} n_i \pmod{q}$.

For each group g , subtracting the mask derived from $S_n^{(t)}$ from the aggregated constant coefficient leaves the sum of the original constant coefficients plus an error term that is a multiple of p . Provided that the resulting integer lies within the centered q -domain decoding interval, reduction modulo p eliminates this error term. The remaining coefficients are lifted from $\mathbb{Z}_p$ to $\mathbb{Z}_q$ without masking and therefore reduce directly to their aggregate values modulo p . Consequently,

$$F_g^{(t)}(X) = \sum_{u_i \in \mathcal{L}_t} f_{(i,g)}(X) \pmod{p}. \quad (77)$$

By linearity of polynomial evaluation, evaluating $F_g^{(t)}$ at the model and auxiliary points yields the corresponding sums of the p -domain masked values. Removing the masks generated from $S_b^{(t)}$, followed by centered lifting and rounding, therefore recovers

$$\widehat{w}_t = \sum_{u_i \in \mathcal{L}_t} w_i, \quad \widehat{\mathbf{y}}_t = \sum_{u_i \in \mathcal{L}_t} w_i \mathbf{x}_{i,t}. \quad (78)$$

Hence, $\widehat{\mathbf{x}}_t = \widehat{\mathbf{y}}_t / \widehat{w}_t$, which proves (76). $\square$

For correct p -domain decoding, the accumulated HPRG errors must satisfy, for every model coordinate ℓ ,

$$\left| \sum_{u_i \in \mathcal{L}_t} e_{i,\ell,t}^{(b)} \right| < \frac{\Delta}{2}, \quad \left| \sum_{u_i \in \mathcal{L}_t} e_{i,t}^{(w)} \right| < \frac{\Delta}{2}. \quad (79)$$

Moreover, since $|\mathcal{L}_t| \leq n'$ and $w_i \leq w_{\max}$, the condition $n'w_{\max} < \tau$ prevents wraparound of the aggregate bucketed

weight. The bounds in (1) provide the required decoding headroom for the scaled p -domain values and for the p -multiple error in the q -domain coefficient layer [11]. Correctness further requires the encoded weighted model coordinates and their active set sums to remain within the prescribed plaintext decoding range, so that reduction modulo τ does not alter the intended values.

B. Data Privacy

Let $\mathcal{H}_t = \mathcal{L}_t \setminus \mathcal{C}$, where $\mathcal{C}$ is the corrupted set and $|\mathcal{C}| < T$. Corrupted users follow the protocol but reveal their local states to the server. For an accepted round t , we consider a left or right privacy experiment in which the adversary chooses two honest input collections with identical aggregate leakage:

$$\begin{aligned} \sum_{u_i \in \mathcal{H}_t} w_i^{(0)} \mathbf{x}_{i,t}^{(0)} &= \sum_{u_i \in \mathcal{H}_t} w_i^{(1)} \mathbf{x}_{i,t}^{(1)}, \\ \sum_{u_i \in \mathcal{H}_t} w_i^{(0)} &= \sum_{u_i \in \mathcal{H}_t} w_i^{(1)}. \end{aligned} \quad (80)$$

The challenger samples $b \leftarrow \{0, 1\}$, executes PQ-DVSA with the b -th collection, and gives the adversary the public transcript, bulletin board postings, corrupted users' states, reconstruction messages, and accepted aggregate output. Let $\text{Priv}_{\text{PQ-DVSA}, \mathcal{A}}^b(\lambda)$ denote this experiment, and let b' be the adversary's output. **Theorem 2 (Data Privacy).** Assume that the RLWE masking mechanisms over p and q are pseudorandom in the multiuser setting even when their aggregate masking secrets are revealed [11], [12], and that the adapted MSSS construction is T -private under aggregate reconstruction [13], [14]. Then, for any probabilistic polynomial time (PPT) adversary controlling the server and fewer than T users,

$$\Pr[b' = b] \leq \frac{1}{2} + \text{negl}(\lambda). \quad (81)$$

Proof. Let H_0 denote the real experiment. In H_1 , replace the honest users' RLWE masks over p , used for the weighted updates, bucketed weights, and reserved verification values, by random masks consistent with the reconstructed aggregate $S_b^{(t)}$. By multiuser RLWE pseudorandomness, H_0 and H_1 differ by at most a negligible advantage $\epsilon_p(\lambda)$. Conditioned on the aggregate mask, the resulting individual masked values are independent of the corresponding plaintexts. Since Lagrange interpolation is an invertible linear transformation, their polynomial coefficient representation reveals no additional information. By (80), the permitted aggregate values are identical in both challenge worlds.

In H_2 , replace the masks applied to the constant coefficients over q by random masks consistent with the reconstructed aggregate $S_n^{(t)}$. This changes the adversary's view by at most a negligible $\epsilon_q(\lambda)$. The remaining coefficients are deterministic linear functions of the values already protected in H_1 , and therefore provide no additional information about the challenge bit.

Finally, in H_3 , replace the honest users' MSSS reconstruction view by a simulation given the public parameters, corrupted users' states, and the aggregate secrets $S_b^{(t)}$ and $S_n^{(t)}$. Since $|\mathcal{C}| < T$, MSSS threshold privacy implies that this changes the

view by at most $\epsilon_{\text{MSSS}}(\lambda)$ and reveals no individual honest b_i or n_i .

In H_3 , the adversary's view depends only on public information, corrupted users' states, and the leakage in (80), which is identical for $b = 0$ and $b = 1$. Hence $\Pr[H_3 : b' = b] = 1/2$. By the triangle inequality,

$$\text{Adv}_{\text{PQ-DVSA}}^{\text{priv}}(\mathcal{A}) \leq \epsilon_p(\lambda) + \epsilon_q(\lambda) + \epsilon_{\text{MSSS}}(\lambda) = \text{negl}(\lambda). \quad (82)$$

Therefore, PQ-DVSA hides each honest user's individual update, bucketed weight, and masking secrets beyond the aggregate leakage specified in (80). $\square$

C. Aggregation Verifiability

Theorem 3 (Aggregation Verifiability). Assume that the bulletin board is authenticated, append only, and globally consistent, and that the decoding conditions of Theorem 1 hold. If an honest user accepts round t , then the accepted coefficient vector equals the coefficientwise sum of the authenticated postings of the declared active set $\mathcal{L}_t$, and the recovered aggregate bucketed weight satisfies the reserved verification relation.

Proof. Let $C_{\text{BB}}^{(t)} = [\sum_{u_i \in \mathcal{L}_t} C_i^{\text{pub}}]_q$ denote the coefficientwise sum obtained from the bulletin board. Global consistency ensures that all honest users observe the same authenticated postings, and acceptance requires $C^{(t)} = C_{\text{BB}}^{(t)}$. Therefore, omission, substitution, or modification of any posted coefficient causes rejection.

If the interpolation points $Z_{g,t}$ remain hidden from the server and cannot be disclosed through collusion, as in the setting considered by SWAS [7], the auxiliary weight verification is sufficient for aggregation verification, since the server cannot intentionally alter the model evaluations while preserving the reserved verification relation. PQ-DVSA additionally considers the stronger case in which $Z_{g,t}$ may become known. The coefficientwise bulletin board check then binds the accepted polynomial to the authenticated postings and prevents an undetected polynomial modification.

After reconstruction of $S_b^{(t)}$ and $S_n^{(t)}$, users recover the aggregate polynomials corresponding to $C_{\text{BB}}^{(t)}$. Demasking the ordinary auxiliary values gives $\hat{w}_t = \sum_{u_i \in \mathcal{L}_t} w_i$. For the reserved value,

$$z_R^{(t)} = \sum_{u_i \in \mathcal{L}_t} w_i + \tau \sum_{u_i \in \mathcal{L}_t} e_{i,t}^{(\text{com})},$$

and therefore $[z_R^{(t)}]_\tau = \hat{w}_t$. Any inconsistency in the recovered aggregate weight is thus detected.

Hence, the coefficient check binds the aggregate to the authenticated postings, while the auxiliary check verifies the aggregate bucketed weight. An honest user therefore accepts only an aggregate satisfying both conditions. $\square$

IX. PERFORMANCE EVALUATION

This section evaluates the learning impact of bucketed weighting and the computation, communication, and storage overhead of our proposed PQ-DVSA scheme.

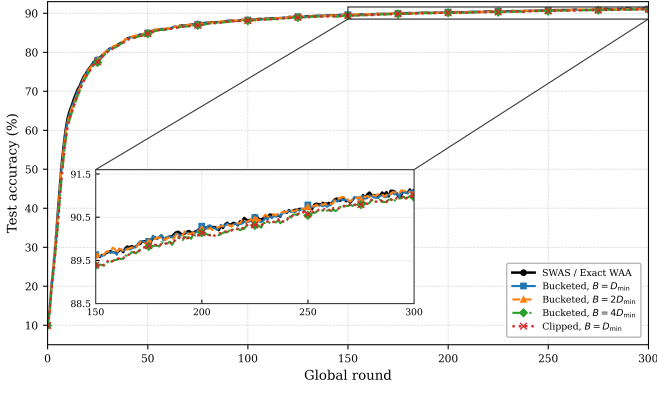


Fig. 1. Test accuracy on Non-IID MNIST for SWAS-style exact weighted aggregation and PQ-DVSA bucketed weighted aggregation.

A. Experimental Setup

For the model accuracy experiment, we isolate the learning effect of bucketed weighting and follow the MNIST/MLP setting used in SWAS [7]. We use the full MNIST training and test sets, an MLP with architecture 784-128-10 and 101,770 trainable parameters, $N = 100$ users, 70% user selection per round, one local epoch, SGD with learning rate 0.001, and batch size 20. The data are split Non-IID using a Dirichlet distribution with $\alpha = 1.0$ and heterogeneous user data sizes.

For the protocol overhead experiments, we measure online operations after local training. Unless otherwise varied, we use $T = 250$, $n' = 260$, $w_{\max} = 2$, and $N_R = 256$, with $N > n'$. We set $\tau = 521$, $p = 211,812,353$, $q = 165,213,724,673$, and $\Delta = \lceil p/\tau \rceil = 406,550$. The primes satisfy $p, q \equiv 1 \pmod{2N_R}$ and the bounds in (1). The parameters M , d , and $G = \lceil d/M \rceil$ are varied in the corresponding benchmarks.

B. Model Performance

Fig. 1 compares exact weight SWAS aggregation with PQ-DVSA bucketed variants under the same Non-IID split, initialization, user selection schedule, and local training procedure. The only difference is the aggregation weight rule. The full plot shows nearly identical convergence, while the zoomed panel highlights the small differences in the final rounds.

The bucket parameters should match the user data size distribution. In our experiments $w_{\min} = 1$, and $D_{\min}$ denotes the minimum user data size. After 300 global rounds, the exact-weight SWAS baseline reaches 91.08% test accuracy. Bucketed settings with $B = D_{\min}$, $B = 2D_{\min}$, and $B = 4D_{\min}$ reach 91.06%, 91.05%, and 90.96%, respectively; the clipped setting with $B = D_{\min}$ and $w_{\max} = 2$ reaches 91.01%. All tested variants therefore remain within 0.12 percentage points of exact weighted aggregation. The average normalized weight deviations are 0.111, 0.176, 0.218, and 0.213, confirming that coarser weights have only a small accuracy effect in this workload.

C. Computational Overhead

The initial TA setup is amortized, while the server mainly performs coefficientwise modular accumulation. We therefore

evaluate the recurring user computation costs of masked submission, MSSS aggregate mask reconstruction, and verification. SWAS [7] is used as the closest functional baseline for verifiable weighted aggregation, while our proposed PQ-DVSA scheme additionally provides post-quantum masking, dropout resilience, and protection under bounded collusion.

1) *User Masked Submission*: Each user masks the d -dimensional weighted update in the p -domain, constructs the auxiliary values, generates $G = \lceil d/M \rceil$ interpolation polynomials, and masks their constant coefficients in the q -domain. Let N_R be the RLWE ring degree and C_η the cost of one RLWE-HPRG ring output over modulus η . The masking cost is

$$T_{\text{mask}}^{\text{PQ}} = \left\lceil \frac{d}{N_R} \right\rceil C_p + \left\lceil \frac{G}{N_R} \right\rceil C_q + O(C_p + d + G). \quad (83)$$

The first term is also the dominant update-masking cost in LHAagg [12]; PQ-DVSA additionally protects the bucketed weight, commitment, and public constant coefficients. With direct Lagrange interpolation,

$$T_{\text{enc}}^{\text{PQ}} = T_{\text{mask}}^{\text{PQ}} + O(M^3 + GM^2). \quad (84)$$

In Fig. 2(a), increasing d with fixed G increases both the update-masking and interpolation costs. In Fig. 2(b), d is fixed and increasing M reduces G , lowering the number of q -domain masks and keeping the PQ-DVSA runtime comparatively stable. For both schemes, $G = \lceil d/M \rceil$. With fixed d , increasing M reduces the number of groups but increases the cost of each direct Lagrange interpolation. When compatible evaluation points are selected, coefficient generation in either scheme may also be accelerated by an INTT; this implementation optimization does not alter the protocol.

2) *MSSS Reconstruction Cost*: After the server publishes the active set and aggregate coefficients, reconstruction users publish MSSS messages for that active set. PQ-DVSA runs two MSSS instances over p and q , and the benchmark measures message generation and recovery of the aggregate masks $S_b^{(t)}$ and $S_n^{(t)}$.

Fig. 3 shows the MSSS runtime as T increases from 250 to 500. The cost grows with T because both message generation and recovery operate on threshold size structures. Recovery is more expensive because it combines at least T valid message pairs. The default PQ-DVSA configuration uses $T = 250$ and $n' = 260$; the larger values $T = 400$ and $T = 500$ are included only as isolated MSSS scalability tests and are not full protocol configurations under this setting.

3) *Bucketed Weight Verification Cost*: To isolate the cost of the weight verification mechanism, we measure verification after the aggregate coefficient vector and active set have been accepted. This includes q domain constant mask removal, auxiliary point evaluation, aggregate masked weight recovery, and commitment verification. The dominant cost is

$$T_{\text{ver}}^{\text{PQ}} = \left\lceil \frac{G}{N_R} \right\rceil C_q + O(GM) + O(G) + O(C_p), \quad (85)$$

where the first term is the q -domain constant-mask removal, $O(GM)$ is the cost of evaluating G degree- M polynomials at the auxiliary points, $O(G)$ covers auxiliary-value processing

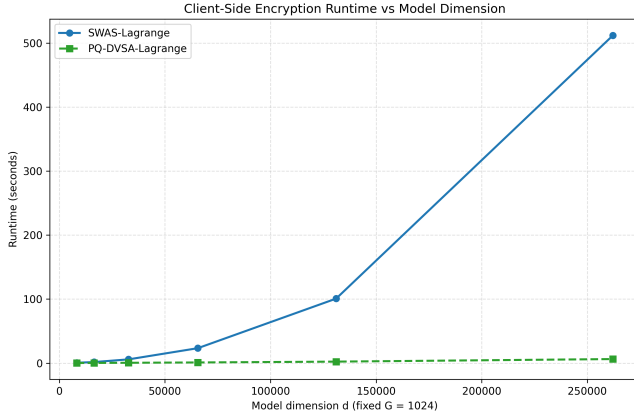
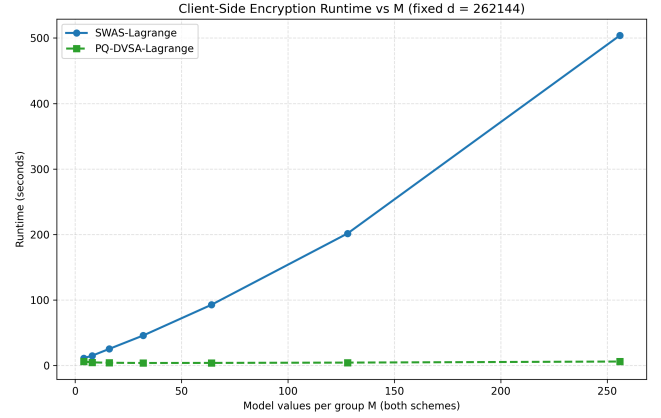
(a) Fixed PQ-DVSA $G = 1024$ (b) Fixed $d = 262144$

Fig. 2. User submission runtime for SWAS and PQ-DVSA using direct Lagrange interpolation. Both schemes encode M model values and one auxiliary verification value per polynomial.

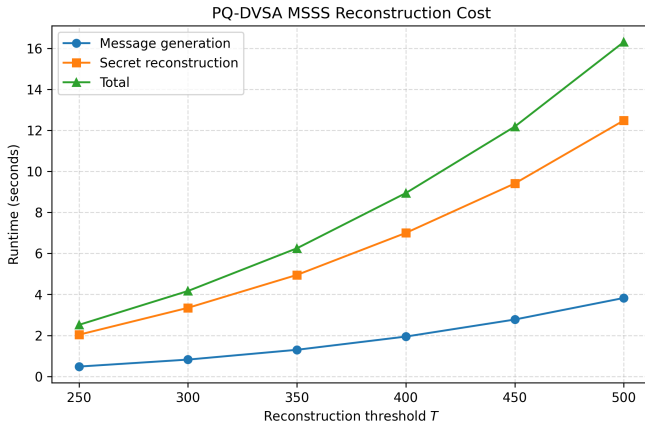


Fig. 3. Client-side MSSS aggregate-mask reconstruction cost in PQ-DVSA as a function of the reconstruction threshold T . A reconstruction subset of size T is used from a larger reconstruction set $\mathcal{R}_t$, where $|\mathcal{R}_t| \geq T$.

and aggregate masked-weight recovery, and $O(C_p)$ accounts for removing the weight and commitment masks.

Fig. 4(a) fixes $M = 64$. Therefore, as d increases, the number of groups $G = \lceil d/M \rceil$ also increases. This increases both the number of q -domain constant masks that must be removed and the number of auxiliary polynomial evaluations, so the PQ-DVSA verification curve grows with d . In Fig. 4(b), $d = 262144$ is fixed. Increasing M reduces $G = \lceil d/M \rceil$, which lowers the number of masked constants and auxiliary values; consequently, the PQ-DVSA verification cost decreases as M grows.

SWAS verification is cheaper because it mainly evaluates auxiliary polynomial values and performs repeated CRT recovery with $k = 16$ residue positions. Its repeated-check cost scales with the number of groups, approximately $\lceil (G_{\text{SWAS}} - 1)/16 \rceil$. Thus, SWAS also increases with d when M is fixed and decreases with M when d is fixed, but its absolute cost remains lower because it does not perform the q -domain RLWE-HPRG mask removal required by PQ-DVSA. The additional PQ-DVSA weight verification cost mainly results from the q domain mask removal and commitment verification required by its protected

polynomial representation.

D. Communication and Storage Overhead

Table III compares the communication overhead of our proposed PQ-DVSA scheme with the relevant baselines using normalized bitlengths rather than a fixed 32-bit representation. Let $\ell_\tau = \lceil \log_2 \tau \rceil$, $\ell_p = \lceil \log_2 p \rceil$, and $\ell_q = \lceil \log_2 q \rceil$ denote the sizes of plaintext entries, p -domain masked values, and q -domain coefficients, respectively. Hence, SWAS and VCD-FL coefficient entries are counted with ℓ_τ , whereas LHagg masked-vector entries are counted with ℓ_p . Since LHagg uses a Shamir-sharing field modulus $P > p$, we write $\ell_P = \lceil \log_2 P \rceil$. We use ℓ_σ and ℓ_{sk} for the signature and signing-key sizes.

In Table III, $G = \lceil d/M \rceil$, where d is the model dimension and M is the group size; T is the reconstruction threshold, n' is the maximum selected or active set size in one round, and N_R is the RLWE ring dimension. For PQ-DVSA, $I_i = \mathbf{1}_{\{u_i \in \mathcal{R}_t\}}$, so only users in the reconstruction set send MSSS messages.

Table III shows the main tradeoff. For the concrete parameters reported below, PQ-DVSA has the largest client upload among the compared schemes because it posts $G(M + 1)$ coefficients with ℓ_q bits per coefficient. This overhead buys public checkability of coefficient aggregation while keeping the encoded updates, bucketed weights, and verification values hidden. In contrast, LHagg uploads only a d -entry masked vector, but dropout recovery requires Shamir multi-secret-sharing traffic proportional to $n'N_R/T$, which is non-negligible for typical RLWE dimensions such as $N_R = 4096$. VCD-FL supports dynamics and collusion-resistant verification, but uses separate commitment material. PQ-DVSA embeds bucketed-weight and commitment checks into the posted coefficient structure, thereby avoiding separate commitment traffic while adding dropout-resilient aggregate-mask recovery over SWAS.

For the concrete parameter set, $\ell_\tau = 10$, $\ell_p = 28$, and $\ell_q = 38$. Table V reports the corresponding PQ-DVSA costs for $M = 64$. The coefficient upload equals $G(M + 1)\ell_q$ bits. The reconstruction columns report $T(\ell_p + \ell_q)$ bits per helper and $T^2(\ell_p + \ell_q)$ bits in total when $|\mathcal{R}_t| = T$.

For storage, Table IV counts only private per-user state. Public parameters, verification keys, interpolation points, MSSS

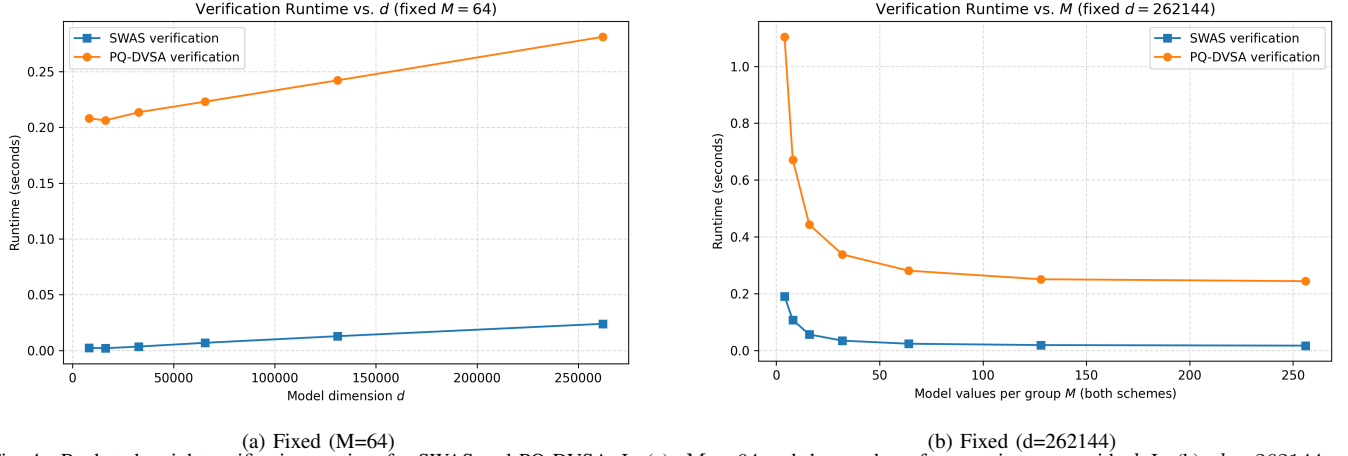


Fig. 4. Bucketed-weight verification runtime for SWAS and PQ-DVSA. In (a), $M = 64$ and the number of groups increases with d . In (b), $d = 262144$ and the number of groups decreases with M . PQ-DVSA includes q -domain constant-mask removal, auxiliary-point evaluation, aggregate masked-weight recovery, and commitment verification. The coefficientwise bulletin-board check and model reconstruction are excluded.

TABLE III
NORMALIZED PER-ROUND COMMUNICATION OVERHEAD.

Scheme	Client Upload	Dropout / Reconstruction	Commitment / Verification
SWAS [7]	$G(M+1)\ell_\tau + \ell_\sigma$	–	embedded in coefficients
VCD-FL [6]	$G(M+1)\ell_\tau$	$T\ell_\rho$ per recovered dropout seed	$GM\ell_\tau + \ell_\tau$, if counted explicitly
LHagg [12]	$d\ell_p$	$\frac{n'NR}{T}\ell_P$ ($+\frac{NR}{T}\ell_P$ for reconstruction)	–
PQ-DVSA	$G(M+1)\ell_q$	$I_i T(\ell_p + \ell_q)$	embedded in posted coefficients

TABLE IV
LONG-TERM PER-USER STORAGE OVERHEAD.

Scheme	Long-Term Per-User Storage	Main Private Material
SWAS [7]	$(N-1)\lambda + \lambda + \ell_{sk}$	pairwise seed row, shared point seed s_z , signing key
VCD-FL [6]	$(N-1)\lambda + N\ell_\rho + \lambda$	pairwise seed row, dropout/share material, verification seed
LHagg [12]	$N_R\ell_p + \frac{NR}{T}\ell_P$	RLWE-HPRG seed and Shamir share material
Our PQ-DVSA	$2\lambda + (L_b + T)\ell_p + (L_n + T)\ell_q$	two local seeds and two MSSS private shares

TABLE V
CONCRETE PQ-DVSA COMMUNICATION COST.

d	G	T	Coeff. Upload (MiB)	Recon. Helper (KiB)	Recon. Total (MiB)
101770	1591	250	0.468	2.01	0.492
101770	1591	300	0.468	2.42	0.708
101770	1591	500	0.468	4.03	1.967
262144	4096	250	1.206	2.01	0.492
262144	4096	300	1.206	2.42	0.708
262144	4096	500	1.206	4.03	1.967

public information, and bulletin-board data are shared public state and are therefore excluded from private storage. Here, N is the total registered-client population, while n' is the maximum selected or active set size per round; in cross-device settings, N is typically much larger than n' .

SWAS and VCD-FL both require pairwise seed rows, together with auxiliary verification or dropout material. LHagg stores RLWE-HPRG seed material and Shamir shares over P . Our PQ-DVSA instead stores two local seeds and the private shares of the two MSSS instances. Its MSSS public parameters

are posted as common bulletin-board data and do not increase private per-client storage.

X. CONCLUSION

This paper presented PQ-DVSA, a post-quantum dropout-resilient verifiable secure aggregation scheme for federated learning with a malicious server and bounded server-client collusion. PQ-DVSA combines RLWE-based masking, two-domain protection, adapted lattice-based MSSS, authenticated coefficient posting, and bucketed-weight verification. Bounded bucketed weights preserve weighted aggregation while limiting the protected weight range, and MSSS reconstructs only active-set aggregate masks without exposing individual masking secrets.

We proved correctness, privacy, and verifiability under bounded collusion, and evaluated model accuracy, computation, communication, and storage overhead. The results show that bucketed weighting has negligible accuracy loss in the tested Non-IID MNIST setting, while the added overhead reflects the cost of post-quantum masking, public coefficient checking, and dropout resilience. Future work includes optimized implementation, larger-scale evaluation, adaptive bucket selection,

post-quantum signatures, and lower-bandwidth coefficient verification.

ACKNOWLEDGMENT

The authors used ChatGPT (OpenAI) for language polishing, structural organization, and LaTeX refinement. The protocol design, security arguments, mathematical derivations, and claims were developed and verified by the authors, who take full responsibility for the manuscript.

REFERENCES

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 2017, pp. 1273–1282. [Online]. Available: <https://proceedings.mlr.press/v54/mcmahan17a.html>
- [2] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 14 774–14 784.
- [3] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1175–1191.
- [4] A. Shamir, "How to share a secret," in *Communications of the ACM*, vol. 22, no. 11, 1979, pp. 612–613.
- [5] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, and J. Zhang, "VFL: A verifiable federated learning with privacy-preserving for big data in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3316–3326, May 2022.
- [6] S. Gao, J. Luo, J. Zhu, X. Dong, and W. Shi, "VCD-FL: Verifiable, collusion-resistant, and dynamic federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3760–3773, 2023.
- [7] Y. He and J. Yu, "Toward secure weighted aggregation for privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 3475–3488, 2025.
- [8] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.
- [9] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '05. New York, NY, USA: Association for Computing Machinery, 2005, pp. 84–93.
- [10] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology – EUROCRYPT 2010*, ser. Lecture Notes in Computer Science, vol. 6110. Berlin, Heidelberg: Springer, 2010, pp. 1–23.
- [11] J. Takeshita, R. Karl, T. Gong, and T. Jung, "SLAP: Simpler, improved private stream aggregation from ring learning with errors," *Journal of Cryptology*, vol. 36, no. 2, p. Art. no. 8, 2023.
- [12] R. Zuo, H. Tian, and F. Zhang, "Post-quantum dropout-resilient aggregation for federated learning via lattice-based PRF," in *Artificial Intelligence Security and Privacy*, ser. Lecture Notes in Computer Science, J. Vaidya, M. Gabbouj, and J. Li, Eds., vol. 14509. Singapore: Springer Nature Singapore, 2024, pp. 382–399.
- [13] X. Zhang, H. Deng, R. Wu, J. Ren, and Y. Ren, "PQSF: Post-quantum secure privacy-preserving federated learning," *Scientific Reports*, vol. 14, no. 1, p. 23553, 2024.
- [14] H. Pilaram and T. Eghlidos, "An efficient lattice based multi-stage secret sharing scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 2–8, Jan. 2017.
- [15] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2020.
- [16] X. Guo, Z. Liu, J. Li, J. Gao, B. Hou, C. Dong, and T. Baker, "VeriFL: Communication-efficient and fast verifiable aggregation for federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1736–1751, 2021.
- [17] X. Zhang, A. Fu, H. Wang, C. Zhou, and Z. Chen, "A privacy-preserving and verifiable federated learning scheme," in *Proceedings of the IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.
- [18] Y. Wang, A. Zhang, S. Wu, and S. Yu, "VOSA: Verifiable and oblivious secure aggregation for privacy-preserving federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 3601–3616, 2023.
- [19] Y. Zhu, J. Gong, K. Zhang *et al.*, "Malicious-resistant non-interactive verifiable aggregation for federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2024.
- [20] W. Yao, T. Zhou, Y. Han, and X. Wang, "Verifiable secure aggregation scheme for privacy protection in federated learning networks," *Discover Computing*, vol. 28, p. 175, 2025.
- [21] Y. Zhang, W. Zhang, Y. Ke, and C. Shen, "VeCoSA-FI: A verifiable and collusion-resistant secure aggregation protocol in federated learning with fair incentive mechanism," *Journal of King Saud University Computer and Information Sciences*, vol. 37, p. 350, 2025.
- [22] D. Becker, J. Guajardo, and K.-H. Zimmermann, "Revisiting private stream aggregation: Lattice-based PSA," in *Proceedings of the Network and Distributed System Security Symposium*, ser. NDSS 2018. Internet Society, 2018.
- [23] S. Yang, Y. Chen, S. Tu, and Z. Yang, "A post-quantum secure aggregation for federated learning," in *Proceedings of the 12th International Conference on Communication and Network Security*, ser. ICCNS '22. Association for Computing Machinery, 2022, pp. 117–124.
- [24] Z. Liu, J. Guo, K.-Y. Lam, and J. Zhao, "Efficient dropout-resilient aggregation for privacy-preserving machine learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1839–1854, 2023.
- [25] Y. Zhang, R. Behnia, I. Karim, A. A. Yavuz, and E. Bertino, "Standing firm in 5G: A single-round, dropout-resilient secure aggregation for federated learning," in *Proceedings of the 18th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '25, 2025, also available as arXiv:2505.07148.
- [26] Y. Zhang, R. Behnia, A. A. Yavuz, R. Ebrahimi, and E. Bertino, "Efficient full-stack private federated deep learning with post-quantum security," arXiv preprint arXiv:2505.05751, 2025.
- [27] X. Qin *et al.*, "Efficient post-quantum cross-silo federated learning," *Mathematics*, vol. 13, no. 9, p. 1404, 2025.
- [28] R. Martínez and P. Morillo, "RLWE-Based Zero-Knowledge Proofs for Linear and Multiplicative Relations," *Cryptology ePrint Archive, Paper 2019/1486*, 2019. [Online]. Available: <https://eprint.iacr.org/2019/1486>