Federated Learning for Binary Neural Networks with Secure Aggregation

Yinuo Wang, Zheng Huang*, Jie Guo, Weidong Qiu
School of Computer Science, Shanghai Jiao Tong University, Shanghai, China
Email: {Yuki123, huang-zheng, guojie, qiuwd}@sjtu.edu.cn

***Abstract*—Federated learning (FL) faces key challenges in communication efficiency and privacy protection. Most existing studies primarily address one of these aspects, while comprehensive security mechanisms often conflict with efficiency requirements. Binary neural networks (BNNs) can substantially reduce computation and communication costs in FL, but aggressive binarization typically leads to severe performance degradation. In this work, we propose *BiSec-FL*, a secure and efficient FL framework that integrates binary aggregation with homomorphic encryption (HE). Specifically, we introduce Scale and Residual Compensated Binary Aggregation (SR-BinAgg) to mitigate quantization loss in binary aggregation, and adopt a selective secure aggregation scheme based on multi-key HE to protect full-precision parameters. Experiments show that BiSec-FL achieves stable convergence across multiple datasets and settings, reducing single-round communication costs by over 96% on average compared with FedAvg, while incurring significantly lower cryptographic overhead than fully encrypted FL systems.**

***Index Terms*—Federated learning, Binary neural networks, Homomorphic encryption**

I. Introduction

Traditional centralized machine learning paradigms require data aggregation, which not only incurs substantial communication and storage overhead but also raises severe privacy concerns. As data protection regulations continue to strengthen worldwide, massive high-value data remains isolated locally and is underutilized. In response, FL has emerged as a promising solution that enables clients to collaboratively train a global model without directly sharing raw data. In a typical FL process, each client performs local training and periodically uploads model updates to a central server for aggregation, thereby enabling collaborative utilization of cross-institutional data across domains such as healthcare, finance, and the Internet of Things (IoT). However, FL systems still face several fundamental challenges in real-world deployments.

First, resource and efficiency constraints pose a major bottleneck. Edge devices typically have limited computation, memory, and communication bandwidth, while FL requires iterative exchange of high-dimensional model parameters. This often results in excessive communication overhead, increased energy consumption, and prolonged training time, especially in large-scale deployments.

*Corresponding author: huang-zheng@sjtu.edu.cn.

This work is supported by the National Natural Science Foundation of China (Grant No. 92270201). The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

Second, privacy risks remain a critical concern. Although FL avoids direct sharing of raw data, prior studies have shown that sensitive information can still be inferred from transmitted parameters during communication and aggregation [1]. Honest-but-curious servers, clients, and eavesdroppers may infer private information from the model updates. While secure aggregation mechanisms can mitigate such risks, applying cryptographic protection to all model parameters usually incurs prohibitive computational and communication costs, limiting their practicality in resource-constrained settings.

Model quantization has been widely employed to improve efficiency by reducing parameter precision. Binarization represents an extreme form of quantization that significantly reduces communication volume and client-side resource consumption when integrated into FL. Beyond efficiency gains, binarization leads to the loss of fine-grained information conveyed by numerical precision, thereby altering the attack surface available to adversaries. This property reshapes the cost structure of secure aggregation and provides new design insights for trading off between efficiency and privacy.

Based on the aforementioned observations, this study systematically investigates the combination of BNNs and secure aggregation techniques in FL. The implementation and experimental scripts are publicly available at https://github.com/Beeaaar/BiSec-FL. Our contributions are summarized as follows:

- We propose a *Binary Secure FL system (BiSec-FL)*. Unlike existing works that treat binarization merely as a compression tool or study secure aggregation as an independent module, BiSec-FL leverages the synergy between the two techniques to jointly achieve communication efficiency and privacy preservation.
- To mitigate the accuracy degradation in binary communication scenarios, we present a lightweight *Scale and Residual Compensated Binary Aggregation (SR-BinAgg)* algorithm, which partially alleviates the accuracy loss caused by aggressive binarization while incurring no additional client-side training cost and only negligible communication overhead. In addition, for mixed-precision parameter transmission, we design a *selective parameter encryption strategy* that significantly reduces cryptographic overhead while protecting critical parameters.
- Through systematic experimental evaluations on multiple datasets under diverse data distributions and client participation rates, we demonstrate that BiSec-FL achieves a fa-

vorable balance among communication efficiency, learning performance, and security. Furthermore, we provide theoretical analyses of communication cost, convergence behavior, and security properties, offering additional support for the effectiveness of the proposed system.

II. Related Work

Existing research on FL has primarily focused on either improving communication efficiency or enhancing privacy protection, while relatively few studies investigate the joint integration of these two objectives to achieve a more balanced and practical system design.

A. Communication Efficiency in FL

Improving communication efficiency has long been a central concern in FL. Early methods such as FedAvg [2] and FedProx [3] reduce communication frequency by performing multiple rounds of local training before global aggregation. Other approaches, including [4] and [5], further reduce system-level communication costs by designing client selection strategies during aggregation. Meanwhile, parameter compression techniques such as FedDQ [6] and FedBDQB [7] reduce the communication volume per round through quantization, while FedMP [8] achieves a similar objective via parameter sparsification. However, most of these studies focus exclusively on improving the communication efficiency, with little attention paid to the training and storage overhead of local devices.

B. Model Quantization and BNNs

To accommodate the constraint of limited local resources, several studies explore low-bit representations during local training, typically by incorporating model quantization into the FL framework. FedQNN [9] investigates the impact of different local quantization levels on FL performance, while FTTQ [10] focuses on ternary quantization for local model training. These works demonstrate that low-bit local training can be practically effective in resource-constrained FL scenarios, such as IoT environments.

As an extreme form of quantization, binary neural networks (BNNs) substantially reduce local computation and storage costs. To compensate for the accuracy degradation caused by binarization, prior studies have proposed various techniques, including quantization error reduction methods such as XNOR-Net [11], improved binarization function designs [12], and network architecture modifications exemplified by BATS [13]. In addition, approaches such as BNN-DL [14] mitigate accuracy loss by introducing specialized constraints into the loss function. These efforts establish important foundations for integrating BNNs into FL. Building on this line of research, BiML-BiFL [15] explores BNN-based FL through a probability-based aggregation strategy, which improves model performance but relies on a sufficiently large client population and incurs increased client-side computational overhead.

C. Privacy-Preserving FL

Meanwhile, privacy-preserving FL has garnered increasing attention, as the update process is vulnerable to security threats such as inference attacks and reconstruction attacks. Differential Privacy (DP) mitigates such risks by injecting calibrated noise into local updated parameters or aggregated models [16], [17], yet it typically incurs certain accuracy degradation. Secure Multi-Party Computation (MPC) enables collaborative aggregation without exposing individual participants' input data [18], among which HE achieves this goal by encrypting model updates and allowing the server to perform computations directly on ciphertext [19], [20].

Although the aforementioned privacy-preserving mechanisms achieve remarkable performance in terms of security, they often introduce considerable computational and communication overhead. Furthermore, most existing approaches adopt a uniform parameter protection strategy, which does not fully account for the efficiency characteristics brought by low-bit or binary numerical representations. In contrast, many high-efficiency FL methods do not explicitly integrate privacy-preserving or secure aggregation mechanisms, leaving potential information leakage risks unaddressed. Consequently, the interplay between extreme model quantization and cryptographic protection mechanisms remains to be thoroughly investigated from a system-level perspective.

III. Preliminaries

Our work operates within a standard Horizontal Federated Learning (HFL) scenario employing a server-client architecture. A central server coordinates N clients to collaboratively train a shared model, with each client holding private local datasets that cannot be directly shared. During an FL round, selected clients perform local optimization and upload their model updates to the server; the server aggregates these updates to generate a global model for the next round.

A. Binary Neural Network Settings

In the local model training process, we adopt BNNs. During forward propagation, binarization is applied to the model weights of convolutional layers and fully connected layers excluding the output layer. Following XNOR-Net [11], we employ the layer-wise scaling factor α to compensate for the magnitude information lost during binarization. Given a real-valued parameter x, its deterministic binarized counterpart is defined as:

$$x^b = \alpha \cdot \mathrm{Sign}(x), \quad \mathrm{Sign}(x) = \begin{cases} +1, & x \geq 0, \\ -1, & \text{otherwise}, \end{cases} \tag{1}$$

where $\alpha = \frac{1}{|x|} \sum_{i=1}^{|x|} |x_i|$. During training, floating-point shadow weights are maintained for optimization. Since the sign function is non-differentiable, gradients are approximated

using the straight-through estimator (STE) based on the Hard-tanh function,

$$\mathrm{Hardtanh}(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \le x \le 1, \\ +1, & x > 1, \end{cases} \tag{2}$$

Gradients are propagated only within bounded intervals, and shadow weights are clipped after each update to ensure training stability. During both model training and inference, sign-based operations in binarized layers replace floating-point multiplications, thereby reducing computational complexity.

B. Threat Model

The security analysis of this work is based on the widely adopted honest-but-curious threat model in privacy-preserving FL. Under this model, both the central server and participating clients adhere to the prescribed training and aggregation protocols, but may attempt to infer private information from accessible model updates or intermediate results. Meanwhile, external eavesdroppers are assumed to be capable of monitoring communications between the server and clients, but lack the ability to perform active attacks such as tampering with communication contents, injecting malicious parameters, or launching Byzantine attacks.

Under the aforementioned threat model, the primary attack scenarios of concern include model reconstruction attacks and membership inference attacks [21], [22]. The security objective of our work is to ensure that the encrypted client model updates satisfy semantic security: during the transmission and aggregation of model updates, these ciphertexts are computationally indistinguishable from encryptions of random messages to any unauthorized party.

IV. The Proposed Framework: BiSec-FL

We propose BiSec-FL, a FL framework that integrates BNNs and HE to jointly address communication efficiency and privacy protection. As illustrated in Fig. 1, BiSec-FL follows the standard server-client FL workflow, while representing and transmitting the majority of model parameters in binary form to reduce communication overhead. To compensate for the expressive accuracy decline induced by binarization, SR-BinAgg is introduced. Meanwhile, a small subset of parameters with full precision are protected using the multi-key HE algorithm xMK-CKKS during upload and aggregation. Parameters of different precisions are aggregated separately at the server and then redistributed to clients for subsequent training rounds, forming an efficient and privacy-preserving FL pipeline.

This section provides a detailed introduction to the binary federated aggregation with scale and residual compensation and selective secure aggregation methods employed in our system. These two methods handle binary and full-precision parameters respectively, jointly ensuring the overall efficiency and security of the system.

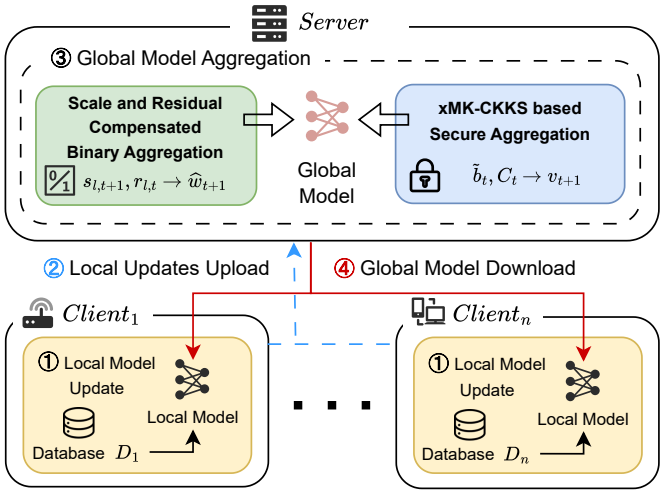

Fig. 1. The framework diagram of proposed BiSec-FL.

A. Binary Federated Aggregation with Scale and Residual Compensation

Although integrating BNNs into FL brings efficiency benefits, directly applying traditional aggregation algorithms such as FedAvg to the binary setting often results in severe accuracy reduction. This performance gap primarily stems from information loss induced by binarization during transmission and aggregation. The method proposed in [15] estimates auxiliary aggregation parameters under a probabilistic framework and partially recover magnitude-related information. However, its effectiveness relies on a sufficiently large number of participating clients per round, an assumption that may not hold in practical FL deployments. To date, no widely accepted solution has effectively bridged the gap between binary federated aggregation and its full-precision counterpart.

In this work, we propose a lightweight compensation strategy termed Scale and Residual Compensated Binary Aggregation (SR-BinAgg) to mitigate the above issue. SR-BinAgg serves as an auxiliary mechanism that can be seamlessly integrated into standard binary federated aggregation frameworks. Importantly, the proposed strategy is unconstrained by the number of participating clients and introduces no additional client-side computation.

In SR-BinAgg (Algorithm 1), n_k denotes the number of local samples held by client k and is used to weight client contributions during aggregation. In each communication round, the server first performs a weighted aggregation over the binarized client updates $\{\hat{w}_{t+1}^k\}$ to obtain an intermediate real-valued representation prior to binarization. Here, $\hat{w}$ denotes a scaled binary model, in which only the sign information represents the binary degrees of freedom, while the magnitude is deterministically specified by a layer-wise scale factor s_l. Furthermore, to alleviate the systematic bias introduced by repeated binarization across communication rounds, SR-BinAgg incorporates a server-side residual accumulation mechanism. Specifically, the server maintains a residual buffer r_l for each

Algorithm 1: Scale and Residual Compensated Binary Aggregation Algorithm (SR-BinAgg)

Server executes:

Initialize residual buffers $\{r_{l,0} = \mathbf{0}\}_l$;
for *communication round* $t = 0, 1, 2, \ldots$ **do**
  Randomly select a subset of m clients S_t;
  foreach $k \in S_t$ **do**
    $(\hat{w}^k_{t+1}, \{s^k_{l,t+1}\}_l, n_k) \leftarrow$ **ClientUpdate**(k, t);
  $n \leftarrow \sum_{k \in S_t} n_k$;
  $\tilde{w}_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} \hat{w}^k_{t+1}$;
  foreach *layer* l **do**
    $s_{l,t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} s^k_{l,t+1}$;
    $\bar{w}_{t+1,l} \leftarrow \tilde{w}_{t+1,l} + \lambda r_{l,t}$;
    $\hat{w}_{t+1,l} \leftarrow s_{l,t+1} \cdot \mathrm{Sign}(\bar{w}_{t+1,l})$;
    $r_{l,t+1} \leftarrow \mu r_{l,t} + (\tilde{w}_{t+1,l} - \mathrm{Sign}(\bar{w}_{t+1,l}))$;
  Broadcast $\hat{w}_{t+1}$ to all $k \in S_t$;

ClientUpdate(k, t)**:**

if $t = 0$ **then**
  Initialize w randomly;
else
  Receive $\hat{w}_t$ from Server;
  $w^k \leftarrow \hat{w}_t$;
for $i = 1$ **to** E **do**
  foreach *mini-batch* $b \in \mathcal{D}_k$ **do**
    $w^k \leftarrow w^k - \eta \nabla_{w^k} \ell(\hat{w}^k; b)$;
    $\hat{w}^k \leftarrow \mathrm{Binarize}(w^k)$;
foreach *layer* l **do**
  $s^k_{l,t+1} \leftarrow \frac{1}{d_l} \sum_{j=1}^{d_l} |w_l^{k,(j)}|$;
return $(\hat{w}^k_{t+1}, \{s^k_{l,t+1}\}_l, n_k)$;

binary layer of the global model, which accumulates the discrepancy between the aggregated real-valued updates and their binarized counterparts. This residual term is injected into subsequent aggregation steps in an error-feedback manner, enabling progressive compensation for binarization-induced errors over time.

The subsequent local optimization procedure follows the same training protocol described in Section III. We define a binarization operator $\mathrm{Binarize}(\cdot)$, which maps floating-point weights to binary weights with layer-wise scaling:

$$\hat{w} = \mathrm{Binarize}(w) = \alpha \cdot \mathrm{Sign}(w), \tag{3}$$

SR-BinAgg introduces only a small number of layer-wise scalar values per communication round, providing a practical solution to narrow the performance gap between binary and full-precision FL while maintaining the communication efficiency of binary joint aggregation. The compensation process involves only binarized parameters, with the remaining few full-precision parameters aggregated using standard FedAvg. This precision-aware parameter separation design also offers flexibility for incorporating security mechanisms.

B. Selective Secure Aggregation with xMK-CKKS

In BiSec-FL, model parameters involved in federated aggregation exhibit heterogeneous sensitivity. The binarized weights, which dominate the communication volume, retain only sign information and therefore expose limited recoverable magnitude information. In contrast, a small subset of full-precision parameters (e.g., bias terms), which are critical for preserving model accuracy, may still carry sensitive information when aggregated jointly with binarized updates. Encrypting all model parameters would provide strong privacy guarantees but incurs prohibitive computational and communication overhead. To balance privacy protection and system efficiency, we adopt a selective secure aggregation strategy that applies cryptographic protection exclusively to full-precision parameters, while allowing binarized weights to be transmitted and aggregated in plaintext.

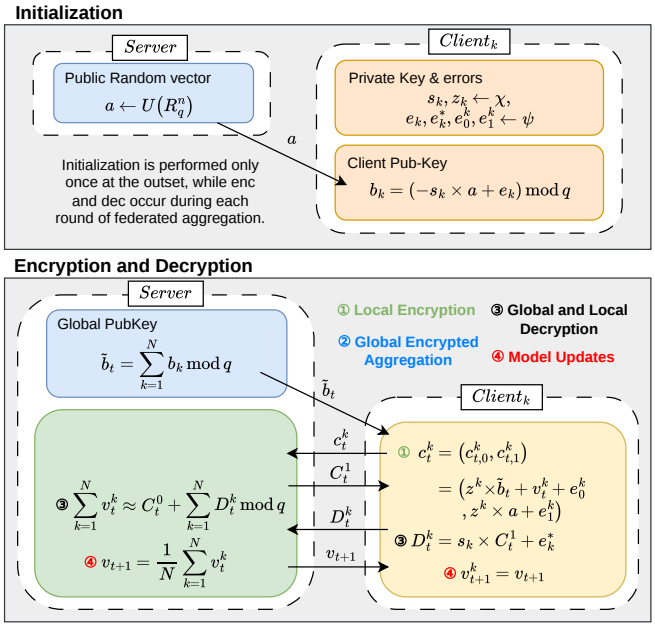

Fig. 2. Selective secure aggregation of full-precision parameters using xMK-CKKS in FL.

Traditional HE schemes rely on shared keys, which, if compromised, can trigger system-wide privacy breaches. Thus we employ a multi-key variant of the CKKS homomorphic encryption scheme, referred to as xMK-CKKS. As illustrated in Fig. 2, during the initialization phase, the server uniformly samples a random common vector a used for public key generation, each client k generates its own public-private key pair (b_k, s_k) and errors from Uniform distribution χ and Gaussian distribution ψ. The private key s_k is retained locally and never revealed, while the public key b_k is uploaded to the central server. The server aggregates all received public keys to construct a collective public key $\tilde{b}_t$ at the beginning of each FL round, which is subsequently used for encryption.

During the client upload phase, full-precision updates are encrypted and encoded as ciphertext c_t^k, while the overall FL

workflow remains unchanged. The server performs ciphertext-domain aggregation by summing the encrypted updates to obtain an aggregated ciphertext C_t. To decrypt the aggregation result, participating clients collaboratively compute partial decryption shares D_t^k using their private keys. These shares are then combined at the server to recover the plaintext sum $\sum_k v_t^k$, ensuring that neither the server nor any individual client can reconstruct private updates on its own. Further cryptographic details of xMK-CKKS can be found in [23].

V. Experiments and Analysis

This section presents both experimental validation and theoretical analysis of the proposed BiSec-FL framework. First, under IID data partitioning of different datasets, we compare the convergence performance of FL methods using only local binary networks, with additional communication-level binarization, and the proposed SR-BinAgg algorithm. Subsequently, we analyze and validate the communication efficiency advantages of BiSec-FL. Furthermore, we evaluate the system's robustness under various data partitioning settings and different client participation rates. Finally, we analyze the security and efficiency benefits achieved by integrating the xMK-CKKS selective encryption mechanism.

A. Experimental Setup

Our primary experiments encompass datasets ranging from simple to complex: MNIST, FEMNIST, and CIFAR-10, employing neural network architectures corresponding to MLP, LeNet, and ResNet-18 respectively. The IID data partition is constructed by uniformly sampling training data across clients, while non-IID settings are evaluated on the FEMNIST dataset, employing label-biased partitioning based on Dirichlet distributions and writer-based data partitioning to reflect the heterogeneity of client data distributions. For BNN implementations, common optimization techniques such as channel widening and layer-wise precision handling are adopted in accordance with the general design principles summarized in [24], without introducing architecture-specific deviations.

Unless otherwise specified, the total number of participating clients and the number of sampled clients per communication round are set to 20 and 5. Each selected client performs local training using stochastic gradient descent (SGD) without momentum and weight decay. Learning rates are selected for different experimental datasets; however, consistent learning rates were maintained across all binary methods under identical experimental settings to ensure fair comparison. Considering the differences in total data volume, the number of local training epochs is 1 for FEMNIST and 3 for the other datasets, with a fixed batch size of 64. For SR-BinAgg, μ and λ are fixed to 0.9 and 0.01. Model performance is primarily evaluated using top-1 classification accuracy on the test set. All experiments are conducted on a Linux-based workstation equipped with an Intel Xeon Silver 4116 CPU and four NVIDIA GeForce RTX 3090 GPUs.

B. Results and Analysis

Specifically, we first present and analyze the overall performance and convergence behavior of all methods on benchmark datasets. Then we demonstrate the communication efficiency advantages of our method, followed by an investigation of model performance under non-IID data distributions and varying client participation rates. Finally, we evaluate the effectiveness of our security mechanisms.

a) Main Results on IID Data and Convergence Analysis: Figure 3 illustrates how test accuracy evolves across FL iterations and communication costs when employing different training and aggregation methods on various datasets. We use centralized training of full-precision networks (FP-Central) and BNNs (Binary-Central) as reference upper bounds. In the FL context, we compared the following approaches: standard FedAvg with full-precision parameters (FP-FedAvg), FedAvg with local binarization only (Binary-Local), fully binarized aggregation with binarization in entire communication process (Binary-FedAvg), and our scale and residual compensation binarized aggregation method (SR-BinAgg).

Results demonstrate that Binary-Local incurs less than 1% final accuracy decline across all three scenarios, indicating that local binarization constitutes a practical choice for resource-constrained clients in FL environments. In contrast, Binary-FedAvg incurs more pronounced accuracy reduction, with performance decline on the MNIST dataset attributable to the simplicity of its MLP architecture. The larger dataset scale of FEMNIST mitigates this phenomenon. On the CIFAR-10 dataset, however, the information loss from aggressive binarization becomes more pronounced due to its inclusion of more complex visual patterns. Across all datasets, SR-BinAgg consistently improves model accuracy compared to Binary-FedAvg, demonstrating its effectiveness in compensating for the accuracy decline caused by aggressive binarization. Notably, in data-rich scenarios, accuracy differs from FP-FedAvg and Binary-Local by merely approximately 0.1%. The lower half of Figure 3 fully demonstrates the system's communication advantages, which will be discussed in the following part.

The convergence rationale of BiSec-FL leverages existing theoretical analyses of FL with BNNs [15] as a qualitative justification, where convergence is established under idealized sufficient conditions (e.g., strong convexity and smoothness of the objective) for optimization with an auxiliary real-valued variable and a sign-based projection onto the binary parameter space. Such analyses demonstrate that, iterative updates followed by binary projection can converge to a neighborhood of an optimal binary solution in theory. Our system is consistent with this projected-iterate optimization structure, therefore also compatible with this analytical framework. Since deriving tight convergence rates for deep, non-convex objectives under arbitrary non-IID data distributions remains challenging, we further validate the convergence behavior of SR-BinAgg empirically in subsequent experiments.

b) Communication Efficiency Analysis: In traditional FedAvg-like algorithms, each client incurs an upload and

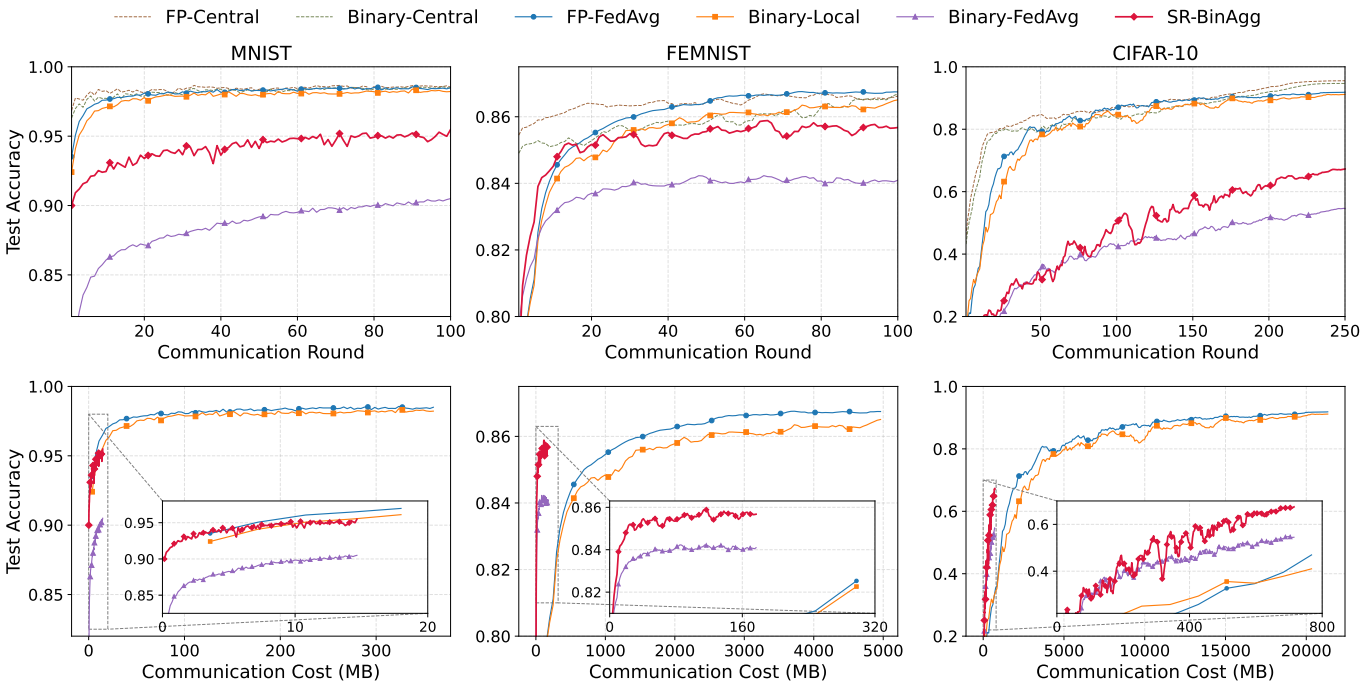

Fig. 3. Test accuracy of different methods over communication rounds and cost on MNIST, FEMNIST, and CIFAR-10 under IID data distributions.

download communication cost of $C_{\text{FedAvg}} = 64P$ bits per round (P denotes the total number of model parameters). For the binary aggregation scheme, model weight parameters are represented using 1 bit, while other parameters retain full precision. Since binary variables constitute an overwhelming majority in common neural networks–exceeding 99% in our models, this approach significantly reduces communication overhead. Compared to Binary-Local, the SR-BinAgg incurs only minimal additional communication overhead, consisting of a small number of full-precision transmitted hierarchical scaling factors. Specifically, the additional relative communication overhead introduced by SR-BinAgg amounts to only 0.0104%, 0.00130%, and 0.00507% on the MNIST/MLP, FEMNIST/LeNet, and CIFAR-10/ResNet-18 datasets, respectively. Thus, the single-round communication cost of SR-BinAgg is practically equivalent to that of Binary-FedAvg in real-world applications.

Table I reports the final test accuracy of the model and the total communication cost per client per round under an IID data distribution. Since all evaluation methods employ symmetric upload and download communication, the reported communication cost corresponds to the total data transmitted by each client during a single FL round. Binary-FedAvg and SR-BinAgg achieve a significant reduction in communication cost by approximately $24\times$ on MNIST and FEMNIST, and over $20\times$ on CIFAR-10. Moreover, SR-BinAgg effectively restores model accuracy at nearly identical communication overhead, achieving an ideal balance between communication efficiency and model accuracy.

Beyond reducing communication overhead, binary weight representation also minimizes local memory usage and computational complexity during forward propagation. Although activation values are currently stored in full precision to ensure training stability, the binary encoding of weights simplifies multiplication operations and alleviates memory bandwidth constraints. Further efficiency gains could be achieved by simultaneously binarizing activations with dedicated hardware support, which is beyond the scope of this work.

c) Performance under Non-IID Data Distributions and Different Client Participation Rate: Figure 4 shows the test accuracy on the FEMNIST dataset under Dirichlet non-IID partitions with concentration parameters $\alpha = 0.1$ and 0.5. Figure 5 reports results under writer-level non-IID partitioning, where $N - K$ denotes the total number of clients and the number of participating clients per round. This setting exhibits stronger statistical heterogeneity. Under such conditions, Binary-Local exhibits severe oscillations and unstable convergence. This occurs because shadow parameters trained by different clients under binary constraints may partially cancel each other out during aggregation. This near-zero aggregation phenomenon triggers frequent sign flips after binarization, leading to instability in the global update process during multiple communication rounds. Since the FedAvg approach does not explicitly address client heterogeneity, model performance exhibits noticeable fluctuations under non-IID data partitions. Meanwhile, SR-BinAgg demonstrates comparable or even superior stability in certain scenarios.

Additionally, we investigate the impact of the number of participating clients per communication round on model performance, as shown in Figure 6. Compared to IID par-

TABLE I

FINAL TEST ACCURACY AND SINGLE-ROUND COMMUNICATION COST PER CLIENT ON IID DATA DISTRIBUTION. ACC. DENOTES TEST ACCURACY (%), AND COMM. DENOTES THE TOTAL UPLOAD AND DOWNLOAD COMMUNICATION COST PER ROUND PER CLIENT (MB).

algorithm	MNIST/MLP		FEMNIST/LeNet		CIFAR-10/ResNet-18	
	accuracy (%)	Comm. (MB)	accuracy (%)	Comm. (MB)	accuracy (%)	Comm. (MB)
FedAvg	98.44 ± 0.05	3.5970	86.73 ± 0.05	49.6160	91.86 ± 0.16	85.3240
Binary-Local	98.24 ± 0.05	3.5970	86.41 ± 0.15	49.6160	91.14 ± 0.20	85.3240
Binary-FedAvg	90.35 ± 0.09	0.1466	84.01 ± 0.18	1.7657	54.59 ± 0.14	2.8598
SR-BinAgg	95.02 ± 0.17	0.1466	85.68 ± 0.16	1.7657	67.26 ± 0.05	2.8600

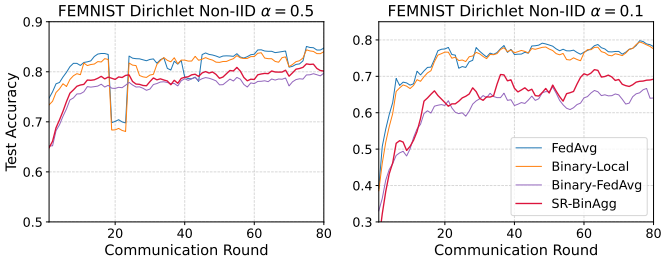

Fig. 4. Test accuracy versus communication rounds on the FEMNIST dataset under Dirichlet non-IID partitions with different heterogeneity levels (α)

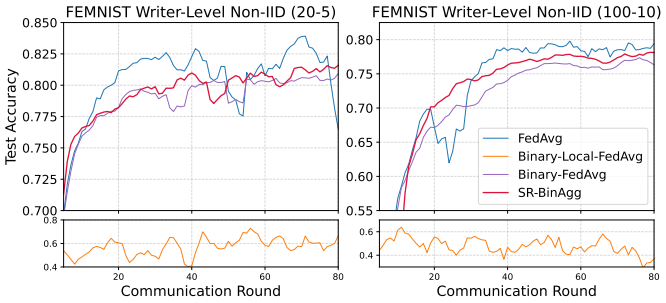

Fig. 5. Test accuracy versus communication rounds on the FEMNIST dataset under writer-level non-IID partitions.

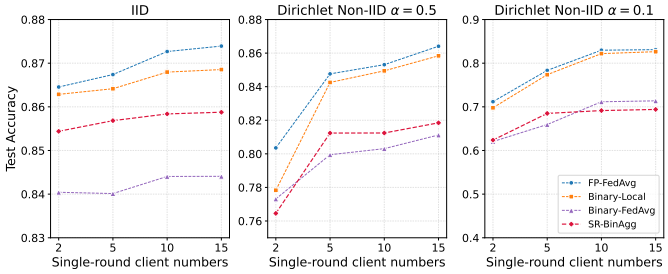

Fig. 6. Test accuracy versus client participation rate on the FEMNIST dataset under different data partitioning strategies.

titioning, under non-IID settings all methods exhibit more pronounced accuracy improvements as the client participation rate increases. However, the performance advantage of SR-BinAgg over Binary-FedAvg is significantly diminished in non-IID scenarios. This is because, under extreme non-IID conditions, conflicting update directions induced by data heterogeneity dominate the optimization process. As SR-BinAgg is primarily designed to compensate for quantization errors rather than mitigate client drift, it behaviorally degenerates toward Binary-FedAvg in such settings. Moreover, excessively large residual momentum μ or compensation strength λ may cause the compensation mechanism to accumulate and propagate heterogeneity-induced noise, thereby undermining global optimization. Therefore, under extreme non-IID settings, we suggest using smaller values of μ and λ to ensure training stability.

d) Security Effectiveness: Table II quantitatively illustrates the efficiency gains of selective encryption across different network architectures on the MNIST dataset. As model complexity increases, encrypting all parameters incurs a rapidly growing computational overhead in both key generation and model aggregation. In contrast, the proposed selective encryption scheme requires only 0.1–0.2 seconds for key generation and significantly reduces the per-round aggregation time. We further evaluate the numerical error introduced by the xMK-CKKS implementation through 5,000 repeated experiments. In practical scenarios where model parameters are concentrated around zero, the relative error is expected to be significantly smaller than 10^{-5}, which is observed under uniformly random plaintext sampling.

TABLE II

KEY GENERATION AND SINGLE-ROUND AGGREGATION TIME COST OF DIFFERENT MODELS ON THE MNIST DATASET UNDER FULL AND SELECTIVE XMK-CKKS ENCRYPTION

Model	Key-gen time (s)		Aggregation time per round (s)	
	Full	Selective	Full	Selective
MLP	4.12	0.08	13.33 ± 0.14	6.33 ± 0.15
LeNet	56.51	0.21	237.05 ± 3.35	52.17 ± 1.16
ResNet-18	135.78	0.19	508.15 ± 6.87	101.16 ± 2.75

BiSec-FL provides heterogeneous privacy protection aligned with parameter precision. Binarization maps real-valued parameters to a highly discrete space, thereby significantly reducing the amount of recoverable information available to gradient-based reconstruction attacks. Prior studies on gradient inversion attacks (e.g., DLG [25]) show that successful reconstruction relies on fine-grained numerical variations, and empirical evidence suggests that sufficiently high compression rates (exceeding 20%) can effectively suppress input data leakage. Binary representations inherently satisfy these defensive conditions. Transmitting binary weights can also effectively protect the system from attacks that directly exploit gradient values [21]. The security of the xMK-CKKS based

HE scheme for full-precision parameters is formally proven in [23], where it is shown that under an honest-but-curious server model, malicious clients, and server-client collusion involving up to $k < N - 1$ clients, individual model updates are computationally indistinguishable from random noise.

VI. Conclusion And Future Work

This study integrates BNNs and HE to achieve an FL system that balances efficiency and security. The proposed lightweight SR-BinAgg and selective encryption mechanisms enable BiSec-FL to effectively mitigate the accuracy degradation caused by aggressive quantization, while preserving the communication efficiency advantages of binary updates. Experimental results demonstrate that BiSec-FL achieves stable convergence under diverse data distributions and varying user participation rates.

Nevertheless, the theoretical analysis of convergence and security in this work primarily builds upon existing results, and rigorous theoretical proofs as well as evaluations against practical attack scenarios are left for future work. Furthermore, addressing the limitations of SR-BinAgg under non-IID data distributions, extending BiSec-FL to stronger adversarial threat models, and exploring adaptive encryption strategies all constitute promising directions for future research toward secure and efficient FL systems.

References

[1] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Curran Associates Inc., 2020.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[4] M. Tang, X. Ning, Y. Wang, J. Sun, Y. Wang, H. Li, and Y. Chen, "Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 102–10 111.

[5] A. M. de Souza, F. Maciel, J. B. da Costa, L. F. Bittencourt, E. Cerqueira, A. A. Loureiro, and L. A. Villas, "Adaptive client selection with personalization for communication efficient federated learning," *Ad Hoc Networks*, vol. 157, p. 103462, 2024.

[6] L. Qu, S. Song, and C.-Y. Tsui, "Feddq: Communication-efficient federated learning with descending quantization," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 281–286.

[7] B. Liu, Y. Gao, C. Zhang, and G. Zhong, "Fedbdqb: Communication efficient federated learning via bidirectional dynamic quantization and bitmap," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2025, pp. 1–6.

[8] B. Isik, F. Pase, D. Gunduz, T. Weissman, and M. Zorzi, "Sparse random networks for communication-efficient federated learning," arXiv:2209.15328[cs.ML], Feb 2023.

[9] Y. Ji and L. Chen, "Fedqnn: A computation–communication-efficient federated learning framework for iot with low-bitwidth neural network quantization," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2494–2507, 2022.

[10] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng, "Ternary compression for communication-efficient federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1162–1176, 2020.

[11] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.

[12] E. Vargas, C. V. Correa, C. Hinojosa, and H. Arguello, "Biper: Binary neural networks using a periodic function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5684–5693.

[13] A. Bulat, B. Martinez, and G. Tzimiropoulos, "Bats: Binary architecture search," in *European conference on computer vision*. Springer, 2020, pp. 309–325.

[14] R. Ding, T.-W. Chin, Z. Liu, and D. Marculescu, "Regularizing activation distribution for training binarized deep networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 408–11 417.

[15] Y. Yang, Z. Zhang, and Q. Yang, "Communication-efficient federated learning with binary neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3836–3850, 2021.

[16] X. Huang, Y. Ding, Z. L. Jiang, S. Qi, X. Wang, and Q. Liao, "Dp-fl: a novel differentially private federated learning framework for the unbalanced data," *World Wide Web*, vol. 23, no. 4, pp. 2529–2545, 2020.

[17] T. Fukami, T. Murata, K. Niwa, and I. Tyou, "Dp-norm: Differential privacy primal-dual algorithm for decentralized federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5783–5797, 2024.

[18] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi, "Privacy preserving vertical federated learning for tree-based models," arXiv:2008.06170[cs.CR], Aug 2020.

[19] N. M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni, and F. Karray, "Secure federated learning with fully homomorphic encryption for iot communications," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4289–4300, 2023.

[20] X. Zhang, A. Fu, H. Wang, C. Zhou, and Z. Chen, "A privacy-preserving and verifiable federated learning scheme," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. Piscataway,NJ: IEEE, 2020, pp. 1–6.

[21] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[22] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[23] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, 2022.

[24] C. Yuan and S. S. Agaian, "A comprehensive review of binary neural network," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 949–13 013, 2023.

[25] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.