A SECURE AND ADAPTABLE PRIVACY-PRESERVING FEDERATED LEARNING FRAMEWORK USING MULTI-KEY FULLY HOMOMORPHIC ENCRYPTION

Neha Umrao [1], Abhay Shukla[2], Somendra tripathi[3]

[1,2,3]Department of CSE, Faculty of Engineering & Technology, Rama University

Email id: neha.umrao11@gmail.com

Abstract:

Federated learning enables collaborative model training without requiring centralized data storage, thereby offering a certain level of privacy protection. However, recent studies indicate that the exchange of model updates, such as gradients or weights, can still lead to unintended leakage of sensitive information. Existing approaches based on single-key homomorphic encryption are insufficient to prevent privacy risks arising from potential collusion between participants and the central server. Moreover, current multi-key homomorphic encryption-based federated learning methods operating in semi-honest environments exhibit limitations in terms of security robustness and practical applicability. To address these challenges, this study introduces a novel privacy-preserving federated learning framework utilizing multi-key fully homomorphic encryption. A new scheme, termed mMFHE, is developed, where encryption is performed through the aggregation of public keys, and decryption requires the collaborative participation of all users. This design ensures enhanced data confidentiality and prevents unauthorized access. The proposed framework secures model updates by applying multi-key fully homomorphic encryption, guaranteeing privacy under the Common Reference String (CRS) model within a semi-honest setting. Additionally, the mMFHE scheme supports both homomorphic addition and multiplication, enabling flexible and advanced computational operations. Security analysis demonstrates that the proposed approach is resilient against collusion attacks involving up to N−1N - 1N−1 users and the server, where NNN represents the total number of participants. Furthermore, experimental evaluations indicate that the scheme reduces the complexity associated with NAND gate operations, thereby lowering computational overhead and improving efficiency, while maintaining high model accuracy.

Keywords: privacy preservation; federated learning; multi-key fully homomorphic encryption

1.Introduction

The conventional development of machine learning models typically depends on the aggregation of distributed datasets into a centralized server or data repository. Although this centralized approach simplifies model training, it introduces several technical challenges, including increased communication overhead and latency associated with large-scale data transmission. More importantly, it raises critical concerns regarding data privacy and security, as sensitive user information must be transferred and stored in a central location. With the rapid proliferation of smartphones and smart devices, an enormous volume of personal data is now generated and stored locally on user devices. Harnessing this data effectively can significantly enhance service quality and user experience, provided that robust privacy protection mechanisms are in place. Federated learning (FL) was proposed as a decentralized paradigm to address these challenges. In this approach, model training is distributed across multiple devices, where each device utilizes its local data to train a model independently. Instead of sharing raw data, only model updates—such as gradients or parameter values—are transmitted to a central server. The server

Doi:10.48047/ijiee.2026.16.2.43

aggregates these updates to construct a global model, which is subsequently shared back with participating devices. By ensuring that raw data remains on local devices, federated learning offers an effective solution for preserving user privacy. Since its introduction, FL has evolved into a rapidly growing research area with applications in diverse domains, including healthcare, the Internet of Things (IoT), and emerging digital environments.

Despite its decentralized nature, federated learning is not entirely immune to privacy threats. Research has demonstrated that model updates exchanged during the training process can inadvertently expose sensitive information about the underlying data. To mitigate these risks, privacy-preserving federated learning (PPFL) has been developed as an extension of FL. PPFL integrates advanced privacy-enhancing techniques such as differential privacy (DP), secure multi-party computation (SMPC), and homomorphic encryption (HE).

Differential privacy safeguards individual data by introducing controlled noise into model parameters or training datasets, thereby limiting the influence of any single data point. However, this noise addition may degrade model performance and slow convergence. Secure multi-party computation enables collaborative computation among multiple participants without revealing their private inputs, but it often incurs substantial computational complexity and communication overhead, particularly in large-scale systems.

Homomorphic encryption presents a promising alternative by allowing computations to be performed directly on encrypted data. This capability enables secure data processing without requiring decryption, thus maintaining confidentiality throughout the computation process. Compared to DP and SMPC, homomorphic encryption can achieve better accuracy while reducing communication overhead. However, traditional single-key homomorphic encryption schemes pose significant security risks, as all participants rely on a shared key. In scenarios where a malicious participant colludes with the central server, the privacy of other users may be compromised.

Existing privacy-preserving federated learning approaches based on multi-key homomorphic encryption (MKHE) attempt to address this issue but still exhibit notable limitations. These schemes often rely on assumptions that restrict their applicability, such as requiring a subset of participants to remain non-collusive. Moreover, many existing solutions support only homomorphic addition and lack support for homomorphic multiplication, thereby limiting their flexibility and applicability in complex machine learning tasks.

To overcome these challenges, this study proposes a novel multi-key fully homomorphic encryption (MKFHE) scheme, referred to as mMFHE. The proposed scheme is designed to enhance privacy protection in federated learning by enabling both homomorphic addition and multiplication operations on encrypted data. It employs an aggregated public key mechanism for encryption and requires collaborative participation among users for decryption, ensuring that no single entity can independently access sensitive information. Additionally, the scheme supports the embedding of multiple plaintext messages within a single ciphertext, improving computational efficiency.

Building upon this encryption scheme, a privacy-preserving federated learning framework is developed. In this framework, model updates are encrypted before transmission and can only be decrypted through collective cooperation among participants. This design significantly enhances resistance to collusion attacks involving both users and the central server. Experimental evaluations conducted using real-world datasets demonstrate that the proposed approach reduces computational complexity particularly in cryptographic

Doi:10.48047/ijiee.2026.16.2.43

operations—while maintaining high model accuracy and efficiency.

The primary contributions of this work can be summarized as follows:

- Development of a novel multi-key fully homomorphic encryption scheme (mMFHE) that ensures enhanced security through aggregated key encryption and collaborative decryption, while supporting efficient multi-bit operations.
- Design of a privacy-preserving federated learning framework based on mMFHE, capable of resisting collusion attacks involving up to N−1N - 1N−1 participants and the server.
- Comprehensive security and performance analysis demonstrating reduced computational overhead and improved efficiency without compromising model accuracy.

The remainder of this paper is organized as follows: Section 2 reviews related work in privacy-preserving federated learning; Section 3 introduces the necessary theoretical background; Section 4 presents the proposed mMFHE scheme and framework; Section 5 provides security analysis; Section 6 discusses performance evaluation and experimental results; Section 7 highlights existing challenges; and Section 8 concludes the study with future research directions.

2. Related Work

2.1 PPFL

Federated learning (FL) is a distributed machine learning framework introduced by McMahan et al. [4] in 2016, which enables multiple participants to collaboratively train a shared model without transferring or centrally storing their local datasets. This approach effectively overcomes the issue of data silos commonly encountered in centralized machine learning systems. Since its introduction, extensive research efforts have been devoted to enhancing FL, focusing on algorithmic improvements, performance optimization, and strengthening security mechanisms.

In the same year, the Federated Averaging (FedAvg) algorithm was proposed [5], establishing a fundamental approach for model aggregation in FL. In this method, each participant independently trains a local model using its private data and periodically transmits updated parameters to a central server. The server aggregates these parameters—typically through averaging—to update the global model, and this process is repeated iteratively until convergence is achieved. Due to its simplicity and effectiveness, FedAvg remains one of the most widely adopted baseline algorithms in federated learning applications.

To address system heterogeneity, Li et al. [16] introduced the FedProx algorithm in 2018, which incorporates a regularization term into the FedAvg framework to stabilize training across diverse devices. Later, in 2020, Asad et al. [17] proposed the FedOpt algorithm, which utilizes advanced optimization techniques, including adaptive gradient-based methods, to improve the efficiency of global model updates and aggregation. In 2023, Zhang et al. [18] developed the Adaptive Locally Aggregated Learning (FedALA) approach, aiming to enhance personalization in federated learning by selectively capturing relevant global information for individual clients. More recently, Yu et al. [19] proposed a Raft consensus protocol based on Cauchy Reed–Solomon (CRS) codes in 2024 for adaptive data maintenance in metaverse environments, effectively reducing storage requirements through erasure coding techniques.

Despite its decentralized design, federated learning is still susceptible to privacy risks. Studies have shown that model updates, such as gradients or weights exchanged during training, can unintentionally reveal sensitive information about the original data, even when the data itself remains on local devices [11,12]. To mitigate such risks, privacy-preserving federated learning (PPFL) has been introduced as an extension of FL, incorporating additional security measures such as encryption and anonymization techniques. Most existing

Doi:10.48047/ijiee.2026.16.2.43

PPFL approaches rely on methods including differential privacy (DP) [20–22], secure multi-party computation (SMPC) [23], and homomorphic encryption (HE) [24,25].

Differential privacy ensures data confidentiality by injecting controlled noise into either the training data or model parameters, thereby limiting the influence of individual data samples on the final output. In 2020, Wei et al. [26] proposed a DP-based framework that enhances privacy protection by adding artificial noise to client-side parameters prior to aggregation. Similarly, Truex et al. [27] incorporated local differential privacy (LDP) into federated learning, introducing the LDP-Fed system, which provides formal privacy guarantees. In 2023, He et al. further advanced this area by proposing ACS-FL, a local differential privacy approach designed to address issues such as variations in parameter ranges across model layers and the excessive accumulation of privacy budgets. Their method employs adaptive cropping, parameter compression, and reorganization techniques to support clustered federated learning on heterogeneous IoT datasets. However, the addition of noise in DP-based methods inherently slows down model convergence and can reduce overall accuracy during aggregation [28].

2.2 HE-Based PPFL

Homomorphic Encryption (HE) provides a powerful capability that allows computations to be carried out directly on encrypted data, eliminating the need for decryption during processing and thereby ensuring data confidentiality throughout the computation process. Leveraging this property, several studies have explored the integration of HE into federated learning to enhance privacy protection.

For instance, Zhang et al. [31] introduced a Privacy-Enhanced Federated Learning (PEFL) framework in which local gradients are encrypted using the Paillier homomorphic encryption scheme before being transmitted to potentially untrusted servers, thereby safeguarding sensitive information. Similarly, Li et al. [32] developed a secure federated learning framework based on the threshold Paillier cryptosystem, particularly targeting Internet of Things (IoT) environments [33], where it helps mitigate risks associated with unreliable or malicious users. In another contribution, He et al. [34] proposed a low-latency, privacy-preserving federated learning scheme designed for edge computing scenarios [35]. Their approach employs an enhanced Paillier encryption mechanism to secure model parameters during transmission, ensuring that raw data remains confined to local devices.

Despite these advancements, most of the aforementioned approaches rely on single-key homomorphic encryption schemes, where all participants share identical encryption and decryption keys. Such a design introduces significant security vulnerabilities, particularly when a malicious participant colludes with the central server, potentially compromising the privacy of other users.

To overcome these limitations and better address privacy requirements in multi-user federated learning systems, researchers have explored multi-key homomorphic encryption (MKHE). Cai et al. [36] proposed a Trusted Execution Environment (TEE)-based MKHE system, known as EMK-BFV, aimed at improving both security and computational efficiency in PPFL. Ma et al. [37] introduced a novel PPFL framework based on the multi-key CKKS scheme, referred to as xMK-CKKS, while Walskaar et al. [38] further enhanced this approach by integrating it with the Flower federated learning framework to improve efficiency. Additionally, Zhang et al. [39] proposed a verification-based PPFL scheme (VPFL) utilizing the BCP cryptosystem, which enables user authentication and ensures data integrity within a multi-key setting.

However, these existing MKHE-based approaches still exhibit notable security and functional limitations. In particular, the

Doi:10.48047/ijiee.2026.16.2.43

schemes presented in [36–38] require that at least two participants remain non-collusive (i.e., k<N−1k < N - 1k<N−1) to preserve privacy in semi-honest environments, meaning that user–server collusion can still pose a threat. Meanwhile, the approach in [39] relies on a dual-server architecture, which assumes that both servers do not collude, thereby introducing additional trust requirements.

Furthermore, limitations also exist in terms of computational functionality. The schemes in [37–39] primarily support homomorphic addition but do not provide full support for homomorphic multiplication. Although the method in [36] enables multiplication, it relies on operations performed within a trusted execution environment, effectively requiring trusted hardware rather than true homomorphic multiplication. As a result, these approaches impose constraints on the types of federated learning algorithms that can be applied and reduce overall system flexibility.

Table 1 provides a comparative overview of existing HE-based PPFL schemes alongside the proposed method, focusing on key aspects such as support for homomorphic multiplication and resilience against collusion attacks between users and servers. Here, kkk denotes the number of colluding users, while NNN represents the total number of participants in the federated learning system.

In contrast, secure multi-party computation allows multiple participants to jointly compute functions without revealing their individual inputs, thereby preserving data confidentiality with minimal impact on model accuracy. Bonawitz et al. [25] were among the first to apply SMPC in federated learning for secure aggregation, utilizing a dual-masking mechanism to protect data during computation. In 2020, Li et al. [29] improved this approach by integrating a single masking strategy with a chained communication mechanism, resulting in a more efficient PPFL framework. Additionally, Gehlhar et al. [30] introduced SafeFL, an SMPC-based framework designed to address both privacy inference attacks and model poisoning threats in federated learning systems. Despite these advantages, SMPC-based methods typically involve complex cryptographic computations and frequent communication among participants, leading to increased computational overhead and latency, particularly in large-scale systems

.

Table 1. Comparison of HE-based PPFL.

Scheme	Base	Homomorphic Addition	Homomorphic Multiplication	Security Against Collusion Attacks
[31,32,34]	Paillier	Yes	No	No
[36]	BFV	Yes	Support for plaintext multiplication in TEE.	$k < N-1$
[37]	MK-CKKS	Yes	No	$k < N-1$
[38]	MK-CKKS	Yes	No	$k < N-1$
[39]	BCP	Yes	No	Requires that the two servers cannot collude
Ours	mMFHE	Yes	Yes	$k = N-1$

3. Preliminaries

3.1. Definitions

For n ∈ N,wedenote the set {1,⋯ ,n} by [n]. For any real number x ∈ R, we define ⌊x⌋ as the greatest integer less than or equal to x, and ⌊x⌉ := ⌊x + 12⌋ as the integer closest to x. Matrices are represented by bold, uppercase letters: A. We use ":=" for deterministic assignments.

Doi:10.48047/ijiee.2026.16.2.43

Definition 1. For a distribution {χn}n∈N based on integers, if it satisfies Pr[|x| ≥ B] = negl(λ),
where negl(λ) is a negligible function, then we call the distribution B-bounded.

Theorem 1. For a range of random variables xi(i ∈ N), if it obeys a B−bounded distribution, then the random variable x = 1 N ∑i=1 xi also obeys the B−bounded distribution.= 1

Definition 2. The statistical distance between two distributions A and B over a finite field Ω is
Δ(X,Y) stat2 ∑t∈Ω | A(t)−B(t) |. A negligible Δ(A,B) implies A state≈ B.

Definition 3 (Learning with Errors, LWE). We consider the case of a secret vector s belonging
to the discrete cube Zn q. The LWE distribution Zn q × Zq over As,χ is established through the
following definition: uniformly sample a ∈ Zn×m qa· s+e)mod q.
and e ← χ,and then output the pair (a,b)

Definition 4 (search.LWEn,q,χ,m). For the secret vector s ∈ Zn q and given m independent samples (ai, bi) ∈ Zn q × Zq to recover s, these are selected from the distributions As,χ. **Definition 5** (Decision.LWEn,q,χ,m). Given m independent samples (ai,bi) ∈ Zn q × Zq, these
samples are selected from the following two distributions: (1) from As,χ; (2) drawn uniformly
from Zn q × Zq. The advantage of being able to distinguish between these two types of selection
is negligible.

Definition 6 (Some-are-errorless LWE). Let q ≥ 1, n > 0, and χ′ be a distribution of errors over
R. Define Tq = {0, 1q ,⋯ , q−1 q }, where q ∈ Z. The distribution A′ s,χ over Tnq × Tq is defined by uniformly selecting a ∈ Tnq and e ← χ′, and outputting (a,b = a · s + e).
The some-are-errorless LWE problem concerns the distinction between two scenarios:

- All samples are uniformly selected from T_{nq} × Tq.
- A random secret vector s ∈ Tnq is uniformly chosen, with the first l samples drawn from A′ s,0 and the remaining samples drawn from A′ s,χ. In other words, the first l samples are of the form (a,b = a·s), which are errorless, while the remaining samples (ai,bi = ai · s + ei) for each i > l introduce a small error e_i

Theorem 2. For any n,l,q ≥ 1(l ≪ n), and an error distribution χ′, there exists a problem ranging from LWEn−1,q,χ to LWEn,q,χ, a variant some-are-errorless problem. LWE polynomials reduce the success advantage of the problem by up to p − n, where p traverses all q prime factors. LWE polynomials, which reduces the success advantage of the problem by at most ∑p p−n, where p iterates over all the prime factors of q. The proof can be found in reference [40]

Definition 7 (Key Homomorphism). Key homomorphism in cryptography refers to a property
of cryptographic algorithms where transformations on keys can be correlated directly with trans
formations on cipher texts [41]. This means that operations performed on keys can have equivalent operations on cipher texts that preserve the structure of the data being encrypted
For example, if there is a cryptographic system that supports key homomorphism, and two keys k1 and k2, performing an operation on these keys (like addition or multiplication) to produce a new key k3 will correlate with a similar operation on cipher texts encrypted with k1 and k2 to produce a new cipher text that would decrypt correctly under k_3.

Doi:10.48047/ijiee.2026.16.2.43

In a system that enables multi-key homomorphism, multiple encryption keys k1, k2,…, can be used simultaneously. When operations such as aggregation or combination (e.g., addition or multiplication) are performed across data encrypted under these different keys, they produce a corresponding new key knew. The key idea is that performing the same operations directly on the cipher texts results in a new encrypted output which, when decrypted using knew, yields a result consistent with the operations applied at the key level. This mechanism allows secure and efficient computation over encrypted data from multiple sources, offering enhanced flexibility for distributed cryptographic applications.

3.2. GSW13

To elucidate the distinctions between our proposed mMFHE scheme and prior works, we provide a detailed exposition of the GSW13. The GSW13, a scheme predicated on the LWE problem, is distinguished by its minimal cipher text expansion during homomorphic operations. We employ the following formal representation for clarity:

Remark 1. Let us consider the positive integers m,m′,n, and q (with $m > n\lceil \log q \rceil$) and a matrix $T \in Zn\times m$. It can be demonstrated that there exists a matrix G, belonging to the set of n × m matrices over the field of integers modulo q, and an inverse function G^{-1}, such that $G^{-1}(T)$ is a binary matrix. Furthermore, it can be demonstrated that the matrix $G^{-1}(T)$ is equal to T. The multiplication of a matrix by G results in a bitwise combination of its elements. In contrast, the inverse of G, denoted G^{-1}, facilitates the bitwise decomposition of these elements. The operational specifics of the GSW13 are outlined as follows:

GSW.Setup $(1\lambda,1d)$: Initiate the setup by defining the lattice dimension $n = n(\lambda,d)$, where λ is the security parameter, and d is an integer specifying the maximum circuit depth permissible. Select a noise distribution $\chi = \chi(\lambda,d)$, bounded by $B\chi$, and determine a modulus q as $q = B\chi 2\omega(d\lambda\log\lambda)$. This configuration is chosen to satisfy the Learning With Errors problem $LWEn^{-1},q,\chi$. Set $m = n\log q+\omega(\log\lambda)$ as the parameter defining the matrix dimensions.

GSW.KeyGen: Generate a uniform random matrix $B \in Z^{(n-1)\times m}$ q and a vector s in Z^{n-1} q . Compute the vector b as b = sB+e, where e is an error vector sampled from a discrete distribution over Zq. The public key is then given by A =key is sk = t = (−s,1) ∈ Z^n q.

GSW.Encrypt: For a plaintext bit message μ, construct a uniform random matrix R $\in\{0,1\}^{m\times m}$, and output the ciphertext C = AR +μG.

GSW.Decrypt: Define the vector w as consisting of zeros, except for the end position set to
⌈q/2⌋. For a given cipher text C, compute $v = tCG^{-1}(w^T) \approx \mu\lceil q/2\rceil$. The decryption yields 0 if v is closer to 0 than to ⌈q/2⌋, and 1 otherwise

.

GSW.Evaluation: Define homomorphic operations as follows:

$$\mathbf{ADD}(\mathbf{C}_1,\mathbf{C}_2)\text{: } Output\ \mathbf{C}_1+\mathbf{C}_2 \in \mathbb{Z}_q^{n\times m}.$$
$$\mathbf{MULT}(\mathbf{C}_1,\mathbf{C}_2)\text{: } Output\ \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)\in\mathbb{Z}_q^{n\times m}.$$
$$\mathbf{NAND}(\mathbf{C}_1,\mathbf{C}_2)\text{: } Output\ \mathbf{G}-\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2).$$

For a more comprehensive analysis and construction details of G and G^{-1}, readers are eferred to additional literature [15].

4. FL Scheme Based on mMFHE

Doi:10.48047/ijiee.2026.16.2.43

This section presents our MKFHE scheme, which is based on the key homomorphism of GSW13 under the CRS model, as well as the multi-bit FHE scheme, which is inspired by GSW13 presented by Li et al. [42], and proves that our scheme also satisfies the key linear homomorphism

4.1. mMFHE

Assuming a CRS model and given the security parameters λ, we set t to the quantity of secret keys and the total number of bits in the message. The i-th participant is designated Pi, i ∈ [N], and N is the number of participants. It is stipulated that each participant has t messages μj ∈ {0,1}, j ∈ [t]. We now give the formal details.

Setting parameters: params ← Setup(1λ,1L):

Setup(·) takes as input the safety parameter λ and the maximum depth L of the circuit. The mode is q = q(λ), the dimension of the lattice n = n(λ), m = m(λ,L) = O(n log q), and the distribution of the errors χ = χ(λ,d) such that (m,n,q,χ) – LWE With the assumption that the security of at least 2λ is achieved against a known attack, a uniformly randomized matrix B ← Zn×m q(as the common string) is chosen. Furthermore, let l = ⌊logq⌋ +1, M = (n+t)·l and output params = (n,q,χ,m), B.

Key generation: (pk,sk) ← KeyGen(params): For the j-th message μj of the i-th par ticipant Pi, select the sample aT j = (aj,1,⋯ ,aj,n) ∈ Z1× n q (Ij | −aT j)T ∈ Z(n+t)× 1 q and output skj := sj = . The important thing to note here is that vj = PowerOf2(sj). (n+t)×t q Most importantly, the private key matrix ski := Si = [sk1,⋯ ,skt] = [s1,⋯ ,st] ∈ Z ; choose ej ← χm×1,j ∈ [t], then calculate bj = B·aj +e(p mod q), and output pki := Ai = [b1 | ⋯ | bt | B] ∈ Zm× (n+t) q , where pk has size O(nm·log2q). Finally, weobserve that A·si = ei and A·S = [e1,⋯ ,et].

Encryption: C ← Enc(params,pk,M): To encrypt a t-bit message, where each bit ujbelongs to the set 0,1 and j ∈ [t], we commence by sampling a uniform matrix Rfrom the set R ← {0,1}m×M. Subsequently, the individual bits of the message areembedded into a diagonal matrix U, expressed as U = diag(u1,⋯ ,ut). This matrix Userves as the basis for constructing the plaintext matrix. The precise method forforming the plaintext matrix will be elaborated as follows:

$$\mathbf{M} = \begin{pmatrix} \mathbf{U}_{t\times t} & \mathbf{0}_{t\times n} \\ \mathbf{0}_{t\times n} & \mathbf{E}_{n\times n} \end{pmatrix} \in \{0,1\}^{(n+t)\times(n+t)}, \quad (1)$$

In this formulation, the matrices U∈Z q t×t and E∈{0,1}n×n are defined as diagonal matrices, where

$$U = \mathrm{diag}(u_1, \ldots, u_t), \quad E = \mathrm{diag}(1, \ldots, 1).$$

These matrices serve as the two partitioning components of the plaintext matrix M. After receiving the public keys from all participants, each participant computes a combined public key given by:

$$PK = A = \frac{1}{N}\sum_{i=1}^{N} A_i.$$

The ciphertext is then constructed as:

Doi:10.48047/ijiee.2026.16.2.43

$$C = M \cdot G + A^T \cdot R \pmod{q}, \quad C \in \mathbb{Z}_q^{(n+t)\times m}.$$

Here, the matrix GGG is defined using the inverse bit decomposition operation:

$$G = \text{BitDecomp}^{-1}(I_{n+t}) = (g^T \otimes I_{n+t}) \in \mathbb{Z}_q^{(n+t)\times(n+t)\cdot l},$$

where I_{n+t} denotes the identity matrix of dimension (n+t). The vector

$$g^T = [2^0, 2^1, \ldots, 2^{l-1}] \in \mathbb{Z}_q^l$$

Evaluation Phase

Upon receiving ciphertexts from all participating entities, each participant performs deterministic homomorphic evaluation based on a predefined circuit .

CIR. The objective is to compute a final ciphertext C, which encodes the result of operations such as addition and multiplication carried out directly on encrypted data

$$\hat{\mathbf{C}} = \mathbf{C}_1 + \mathbf{C}_2 = (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{G} + \mathbf{A}^T(\mathbf{R}_1 + \mathbf{R}_2) \in \mathbb{Z}_q^{(n+t)\times M} \quad (2)$$

Mult (C_1,C_2):Outputting the matrix product,as $C_2=M_2 \cdot G+A^T \cdot R_2$,one obtains

$$\begin{aligned}\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) &= \left(\mathbf{M}_1 \cdot \mathbf{G} + \mathbf{A}^T \cdot \mathbf{R}_1\right) \cdot \mathbf{G}^{-1}(\mathbf{C}_2) = \mathbf{M}_1 \cdot \mathbf{C}_2 + \mathbf{A}^T \cdot \mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \mathbf{M}_1\mathbf{M}_2 \cdot \mathbf{G} + \mathbf{A}^T\mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{A}^T\mathbf{R}_2 \in \mathbb{Z}_q^{(n+t)\times M}\end{aligned} \quad (3)$$

Additionally, this configuration facilitates the computation of homomorphic NAND gates. The operation is executed by generating the output from the expression $G-C_1G^{-1}(C2)$.

4.3. Threat Model In contemporary encryption protocols, it is commonly postulated that participants align with the "honest but curious" model. In accordance with this assumption, partici pants faithfully execute the prescribed protocol while simultaneously seeking any feasible means to derive confidential data contained within the output generated throughout the protocol's execution. In our research, we employ MKFHE to safeguard data privacy within a federated learning framework. Specifically, we posit that both the server and all remote participants operate under the honest-but-curious assumption. This implies that while they conscientiously adhere to the protocol's specifications, they concurrently endeavor to derive personal information about other participants from the shared data during the course of the protocol's execution. Additionally, we entertain the possibility of collusion among the participants and the server. To elucidate, we consider a specific adversarial scenario with k = N −1 participants, where N denotes the total number of participants and k the number of conspirators, collaborating with the server to compromise the confidentiality of a targeted participant. This scenario highlights the potential vulnerabilities and the requisite safeguards necessary in the design of secure federated learning systems.

Doi:10.48047/ijiee.2026.16.2.43

4.4. Our PPFL Scheme Building upon the mMFHE scheme proposed in the preceding section of this doc ument, we introduce a privacy-centric FL scheme. This scheme is predicated on the assumption that all participants actively engage and contribute to the model training process in each iteration. With reference to the FedAvg algorithm, the general process of our PPFL scheme is illustrated in Figure 1. In our proposed PPFL scheme, based on the mMFHE scheme and in referencing the FedAvg algorithm, the process is outlined as shown in Figure 1. During each model aggregation round, clients train the received global model using their local data. Clients independently decide the number of training rounds and parameters based on their resources and data volume. After training, clients encrypt their model parameters using aggregated public key PK and send the encrypted results back to the central server.

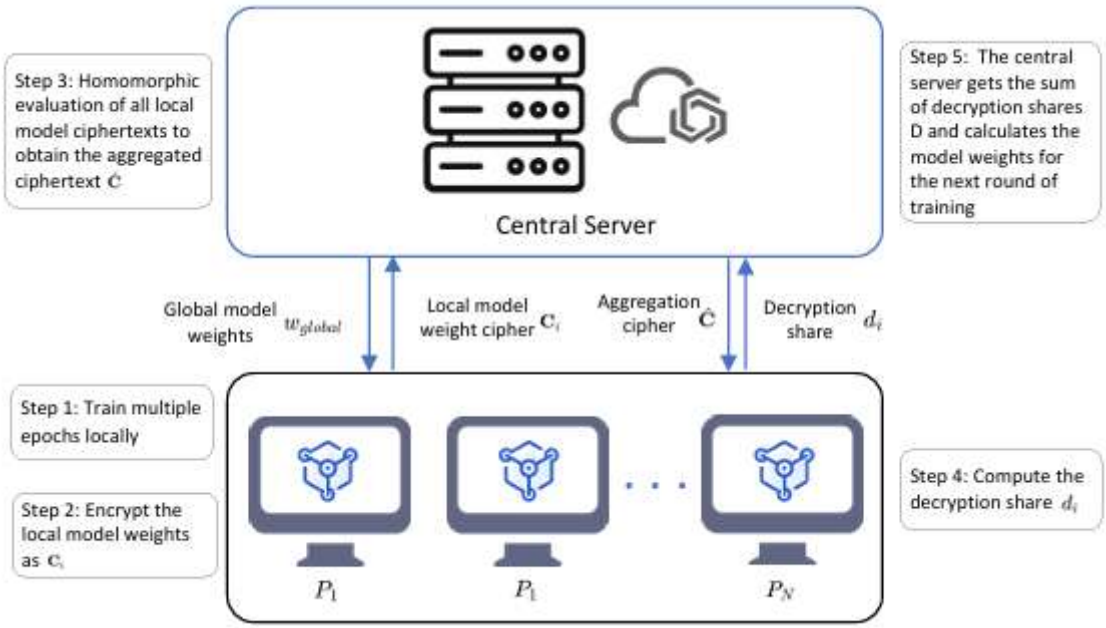

Figure 1. Model of mMFHE-based PPFL scheme.

The central server collects all encrypted model parameters from the clients and per forms a ciphertext aggregation to obtain the aggregated result ciphertext ˆ C. Then, clients use their private key matrices ski and the ˆ C from the server to compute decryption shares di and send these shares back to the central server. The server aggregates all decryption shares to obtain the plaintext form of the model aggregation result and updates the global model using the average aggregation method. This process is repeated until the model meets the predetermined performance standards or completes the specified number of training rounds. The aforementioned flow of encryption and decryption is illustrated in Figures 2 and 3, and a detailed description of each step is provided below

.

Doi:10.48047/ijiee.2026.16.2.43

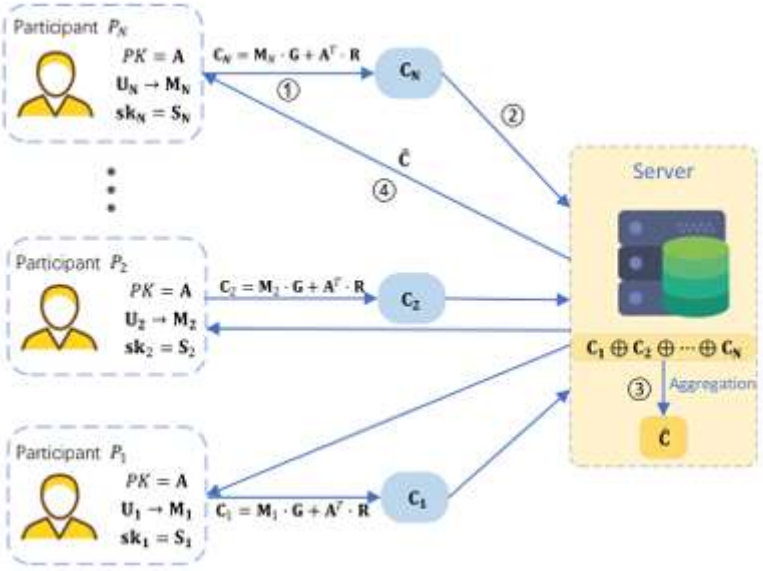

Figure 2. Encryption process

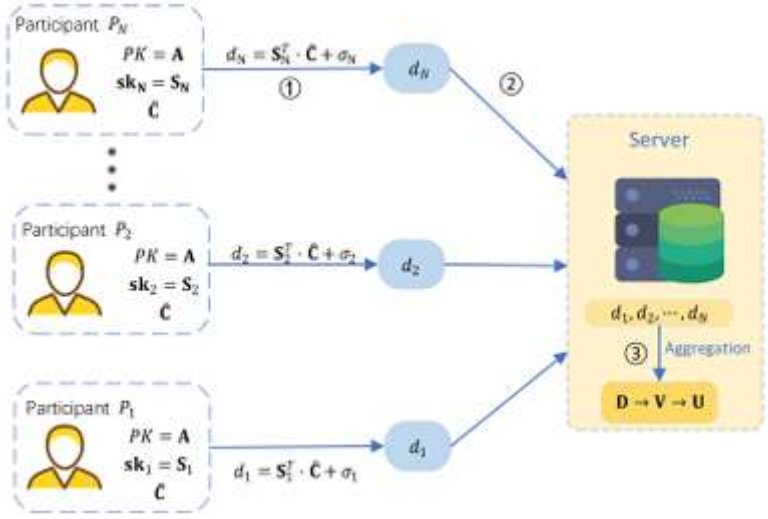

Figure 3. Decryption process.

Initialization: The central server performs the initialization of the global model; executes the Setup (·) function; sets the ciphertext mode q, the lattice dimensions n, m, the error distribution χ, and the length of the message t according to λ and L; selects the uniform random matrix B ← Zn×m q ; and returns public parameters params = (n,q,χ,m), public matrix B, and t. Each remote client Pi (i ∈ [N]) selects the sample aT j = (aj,1,⋯ ,aj,n) ∈ Z1×n in order to generate the private key matrix:

$$\mathbf{sk}_j := \boldsymbol{S}_i = [\mathrm{sk}_1, \cdots, \mathrm{sk}_t] = [\mathbf{s}_1, \cdots, \mathbf{s}_t] \in \mathbb{Z}_q^{(n+t)\times t} \tag{9}$$

where $\mathrm{sk}_j := \mathrm{sj} = (\mathrm{Ij} \mid -\mathrm{aT\ j})\mathrm{T} \in \mathrm{Z(n+t)\times\ 1\ q}$ (9) is the j^{-th} bit of the message corresponding to the key, j ∈ [t]. Subsequently, select ej ← χ m×1 and compute bj = B·aj +ep mod q to generate and return the public key matrix to the centralized server:

$$\mathbf{pk}_i := \boldsymbol{A}_i = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_t \mid \mathbf{B}] \in \mathbb{Z}_q^{m\times(n+t)} \tag{10}$$

The central server receives the public key matrix from all participating clients, calculates

Doi:10.48047/ijiee.2026.16.2.43

$$PK = \mathbf{A} = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{11}$$

and returns the aggregated public key PK to all participants (clients).

Step 1: Local Training: At the beginning of each aggregation round, each participant Pi receives the global model weights w global from the central server. They then utilize their own locally held data to train the model, and after enough epochs, the participant Pi generates a locally held model with the weights wi.

Step 2: Model Weight Encryption: Let the t-bit message ui ∈ {0,1} be the encoded plaintext input of wi and generate the corresponding plaintext matrix Mi (see Section 4.1 above); participant Pi samples a homogeneous matrix Ri ← {0,1}m×M and encrypts Mi with the aggregation public key PK = A to obtain the ciphertext

$$\mathbf{C}_i = \mathbf{M}_i \cdot \mathbf{G} + \mathbf{A}^T \cdot \mathbf{R} \pmod{q} \in \mathbb{Z}_q^{(n+t)\times M} \tag{12}$$

and sends Ci to the central server.

Step 3: Homomorphic Evaluation: The central server performs homomorphic evaluation on the cipher texts after receiving the cipher texts with model weights sent by each parti ci pant. For ease of understanding, here in this paper, there is an example of homomorphic addition, which yields

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} \mathbf{C}_i = \sum_{i=1}^{N} \mathbf{M}_i\mathbf{G} + \mathbf{A}^T \sum_{i=1}^{N} \mathbf{R}_i \in \mathbb{Z}_q^{(n+t)\times N}; \tag{13}$$

The server then sends ˆ C to all participants. And if we want to perform homomorphic multiplication, in the case of two participants,

$$\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) = \mathbf{M}_1\mathbf{M}_2 \cdot \mathbf{G} + \mathbf{A}^T\mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{A}^T\mathbf{R}_2 \in \mathbb{Z}_q^{(n+t)\times M} \tag{14}$$

Step 4: Calculation of decryption share: In MKFHE, the decryption of the ciphertext necessitates the input of all participating members; in order to decrypt ˆ C, it is necessary for each participant Pi to calculate its decryption share di using its private key matrix Si:

$$d_i = \mathbf{S}_i^T \cdot \hat{\mathbf{C}} + \sigma_i \in \mathbb{Z}_q^{(n+t)\times t} \tag{15}$$

where σi = σ′ 1,...,σ′ j ,...,σ′ t ∈ Z(n+t)× t q is the matrix of random vectors, and σ′ j is the error vector chosen from χ.

5. Security Analysis

This section presents a discussion of the manner in which the presented scheme ensures the confidentiality of the model weights within the FL system. This, in turn, ensures the privacy of the data hosted on the distributed devices by each FL participant. In order to characterize the security of our scheme for each potential adversary in the CRS model, we will employ the following theorem.

Theorem5 (Semantic Safety). mMFHE is IND-CPA-safe if the parameters params = (n,q,χ,m,t) are chosen to align with the difficulty presumption of the LWEn,m,q,χ problem, and m = O(n logq).

Doi:10.48047/ijiee.2026.16.2.43

6. Performance Analysis and Experimentation

6.1. Performance Analysis

This section presents an analysis of the performance of the proposed mMFHE protocol and makes some comparisons with existing related schemes. mMFHE is implemented based on the GSW13 scheme. Since homomorphic NAND is realized through a combination of homomorphic addition and homomorphic multiplication, we can evaluate its efficiency by analyzing the complexity of NAND. In comparison to other FHE schemes, the mMFHE scheme exhibits a time complexity of ˜O(N(nd)ω) for evaluating NAND gates, where n represents the lattice dimension, d denotes the depth of the NAND circuit being evaluated, and ω < 2.3727 is a fixed constant [44]. In parallel research, detailed in reference [45], another LWE-based FHE scheme, referred to as Bv11, serves as the foundation for constructing SMC protocols via threshold decryption. The Bv11 scheme achieves a complexity of ˜O(n3d) for evaluating NAND gates. This positions it as marginally less efficient than mMFHE, particularly when the lattice dimension n is large. More importantly, the cipher texts in mMFHE are in matrix form. This implies a lower expansion rate for ciphertext size, reduced time consumption, and the ability to avoid evaluation key operations in homomorphic evaluation, which are often the most time- and space-consuming aspects of FHE and related applications. In recent years, several FHE schemesbasedonGSW13 have emerged, such as in[15,40]. These schemes follow "matrix cascading" approach to construction in cipher texts, resulting in large ciphertext sizes and computational assumptions. Moreover, the aforementioned schemes are single-bit; if a participant's input is t bits, the two schemes need to be executed t times. In contrast, the scheme proposed in this paper only requires a single execution, making it more time-efficient than existing schemes. The details are shown in Table 2, where "CTE Ratio" represents the ciphertext expansion ratio, n is the lattice dimension, N is the number of users, EK indicates whether the key needs to be evaluated, and "NAND TCP" is the time complexity consumed by each NAND gate

Table2.Comparison of computational performance of related FHE schemes.

Scheme	Base	CTE Ratio	EK	NAND TCP
[15]	GSW13	$O(1)$	No	$\tilde{O}(tN(nd)^{w})$
[46]	NTRU	$O(1)$	Yes	Depend on N
[45]	Bv11	$(n+1)\log q$	Yes	$\tilde{O}(n^{3}d)$
[40]	GSW13	$O(1)$	No	$\tilde{O}(t(nd)^{w})$
Ours	GSW13	$O(1)$	No	$\tilde{O}((nd)^{w})$

6.2 Experimentation and Evaluation

In this study, the performance of the proposed **mMFHE** and **mMFHE-based Privacy-Preserving Federated Learning (PPFL)** schemes was systematically tested and evaluated. Furthermore, a set of comparative experiments was conducted to validate and benchmark the effectiveness of the proposed approach against existing methods.

Simulation Environment and Experimental Setup

The implementation and evaluation were carried out under the following configuration:

- **Hardware Configuration:**
  A system equipped with a 13th Gen Intel® Core™ i9-13900HX processor (2.20 GHz) and 32 GB RAM was used.
- **Operating System:**
  Ubuntu Server 18.04 LTS was deployed on the server side, while Ubuntu 18.04 LTS was used for user-side operations.

Doi:10.48047/ijiee.2026.16.2.43

- **Datasets Used:**
  The model was trained and evaluated using three widely recognized benchmark datasets:
  - FEMNIST
  - EMNIST
  - Fashion-MNIST
- **Model Architecture:**
  A neural network model with a fully connected structure (CNN-based configuration) was employed.
  - **Input Layer:** 784 neurons
  - **Hidden Layers:** Five-layer architecture with ReLU activation function
  - **Output Layer:** Softmax activation function for classification
- **Training Configuration:**
  - Optimizer: Adam optimizer
  - Learning Rate: 0.01
  - Local Training: 20 epochs per communication round
- **Security Parameter:**
  The cryptographic parameter was set as:

$$q = 2^{31} - 1$$

Table3. Datasets.

Scheme	Base	CTE Ratio	EK	NAND TCP
[15]	GSW13	$O(1)$	No	$\tilde{O}(tN(nd)^w)$
[46]	NTRU	$O(1)$	Yes	Depend on N
[45]	Bv11	$(n+1)\log q$	Yes	$\tilde{O}(n^3 d)$
[40]	GSW13	$O(1)$	No	$\tilde{O}(t(nd)^w)$
Ours	GSW13	$O(1)$	No	$\tilde{O}((nd)^w)$

Comparative Evaluation of Proposed Scheme

To validate the effectiveness of the proposed approach, a comprehensive comparison was conducted between **mMFHE-based federated learning** and three alternative frameworks, namely **MK-CKKS-based FL**, **TEE-based FL**, and traditional federated learning. The evaluation focused on key performance indicators, including computational cost, memory overhead, communication cost, and model accuracy. Traditional federated learning does not incorporate additional privacy-preserving mechanisms for model updates, making it vulnerable to potential data leakage. In contrast, MK-CKKS-based federated learning ensures confidentiality by encrypting model updates using the multi-key CKKS scheme; however, it introduces certain privacy risks due to the reliance on noise flooding techniques during decryption.

In terms of computational cost, the proposed mMFHE scheme was compared with MK-CKKS across fundamental operations such as homomorphic addition, homomorphic multiplication, encryption, and decryption. Although both schemes exhibit comparable performance in encryption and decryption phases, mMFHE demonstrates overall computational advantages. Specifically, the execution time for homomorphic addition in mMFHE increases linearly with the number of operations and consistently outperforms MK-CKKS. For homomorphic multiplication, MK-CKKS initially shows slightly better efficiency; however, as the number of multiplications increases, its performance degrades due to the need for complex operations such as noise rescaling. Consequently, mMFHE achieves superior

Doi:10.48047/ijiee.2026.16.2.43

efficiency in large-scale multiplication tasks.

Regarding memory consumption, TEE-based federated learning transfers the entire privacy-preserving process to secure memory environments, whereas both mMFHE and MK-CKKS operate using multi-key homomorphic encryption within standard memory architectures. Experimental observations indicate that mMFHE maintains a balanced memory usage while ensuring strong privacy guarantees, making it a more practical and scalable solution compared to alternative approaches.

Overall, the results highlight that the proposed mMFHE-based federated learning framework achieves an effective balance between computational efficiency, memory utilization, and privacy preservation, thereby demonstrating its superiority over existing methods in distributed secure learning environments. The memory overhead of the three approaches was carefully analyzed and compared. Due to its relatively lower ciphertext expansion rate, the memory consumption of the mMFHE scheme lies between the other two methods. Specifically, mMFHE requires more memory than the secure memory utilized in TEE-based federated learning, but significantly less than the conventional memory usage observed in MK-CKKS-based federated learning.

Although TEE-based FL demonstrates lower memory overhead, it suffers from critical security limitations, particularly in scenarios involving collusion between users and servers. Furthermore, TEE environments rely on costly page-swapping mechanisms within secure memory, which can negatively impact system efficiency.

Considering these factors, the slightly higher memory overhead of mMFHE is justified, as it provides stronger security guarantees without incurring excessive resource consumption, thereby representing a practical and balanced trade-off between efficiency and privacy.

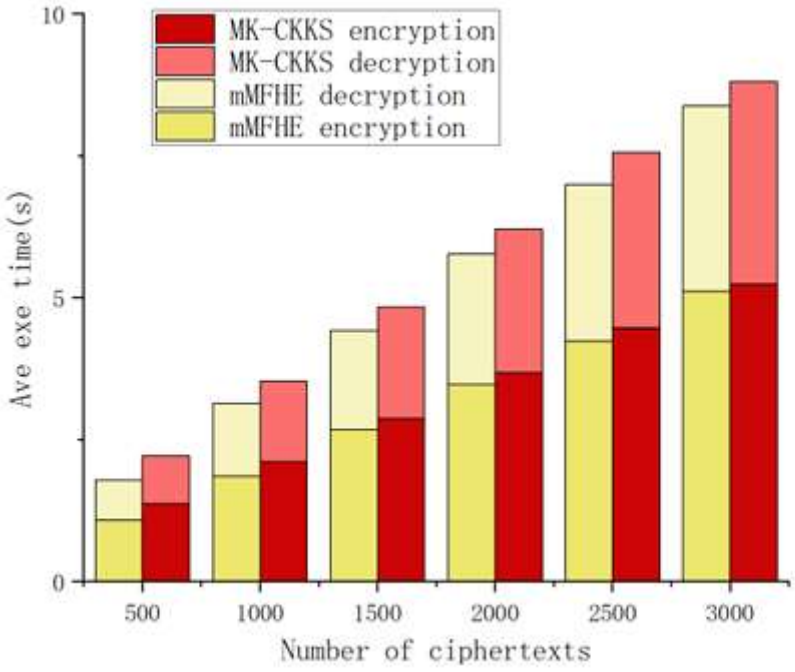

Figure 4. Encryption and decryption.

Doi:10.48047/ijiee.2026.16.2.43

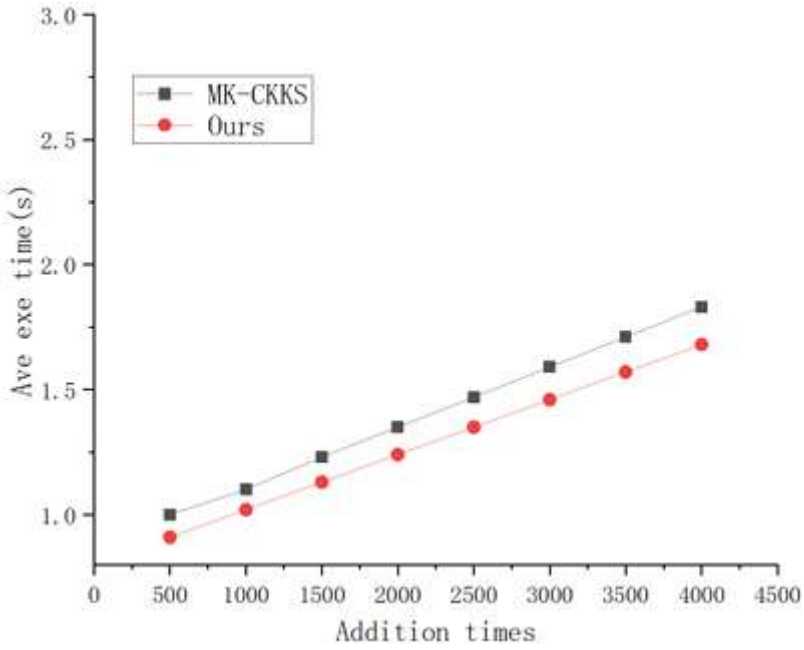

Figure 5. Addition.

Communication Overhead and Accuracy Analysis

The communication overhead of the proposed mMFHE-based PPFL scheme was further analyzed by measuring the total data exchanged during a single training round. In this framework, the primary data transmitted from users to the server consists of encrypted model updates. The communication cost was evaluated for varying model sizes ranging from 10410^4104 to 10610^6106 parameters. As reported in Table 4, both ciphertext size and communication requirements increase with the number of model parameters. Notably, when the model size reached 7,027,860 parameters, the total communication overhead was approximately **32.2 MB**, which remains within a practical and acceptable range for real-world distributed learning systems.

In terms of model performance, accuracy was assessed as the proportion of correct predictions made by the global model on the test dataset. The accuracy of the mMFHE-based PPFL scheme was compared with traditional federated learning and MK-CKKS-based FL. The results indicate that when the number of local training epochs is increased to L=40L = 40L=40, the proposed scheme achieves an accuracy of **97.1%**, which is very close to that of traditional federated learning (**97.9%**) and notably higher than that of the MK-CKKS-based approach. This improvement is primarily attributed to the fact that MK-CKKS relies on approximate arithmetic, leading to cumulative errors during encryption and computation. In contrast, the mMFHE scheme ensures more precise decryption, thereby maintaining higher model accuracy while preserving data privacy.

Overall, the findings demonstrate that the proposed approach effectively balances communication efficiency and predictive performance, making it a reliable solution for privacy-preserving federated learning systems.

Doi:10.48047/ijiee.2026.16.2.43

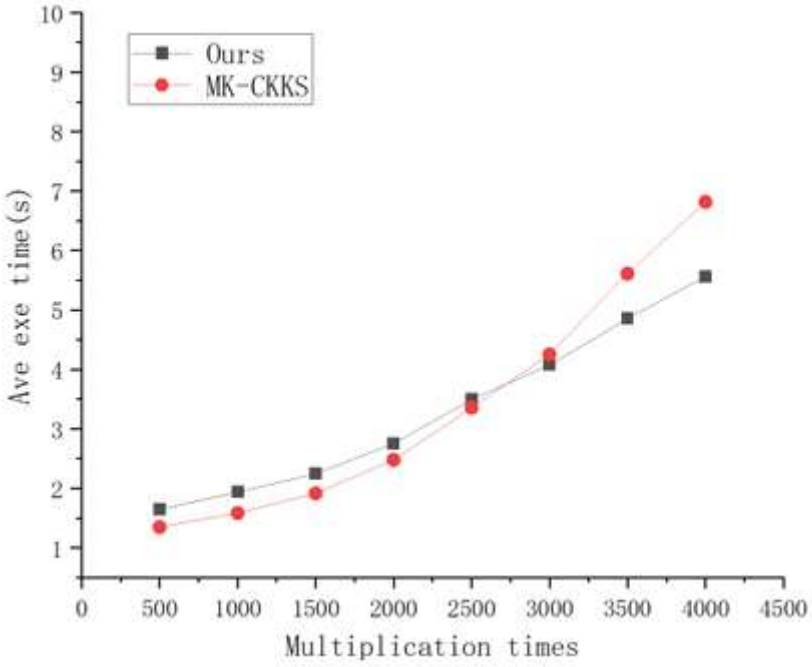

Figure 6. Multiplication.

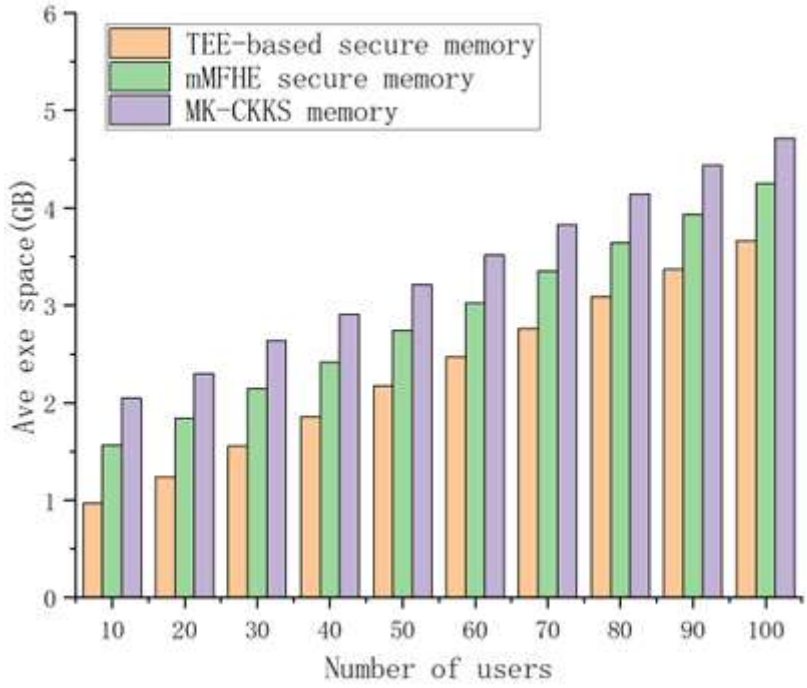

Figure 7. Memory consumption.

Table 4. Communication overhead

Model Size	Ciphertexts Num	Communication Overhead
50,670	63	232 KB
616,420	762	2.8 MB
7,027,860	8660	32.2 MB

Doi:10.48047/ijiee.2026.16.2.43

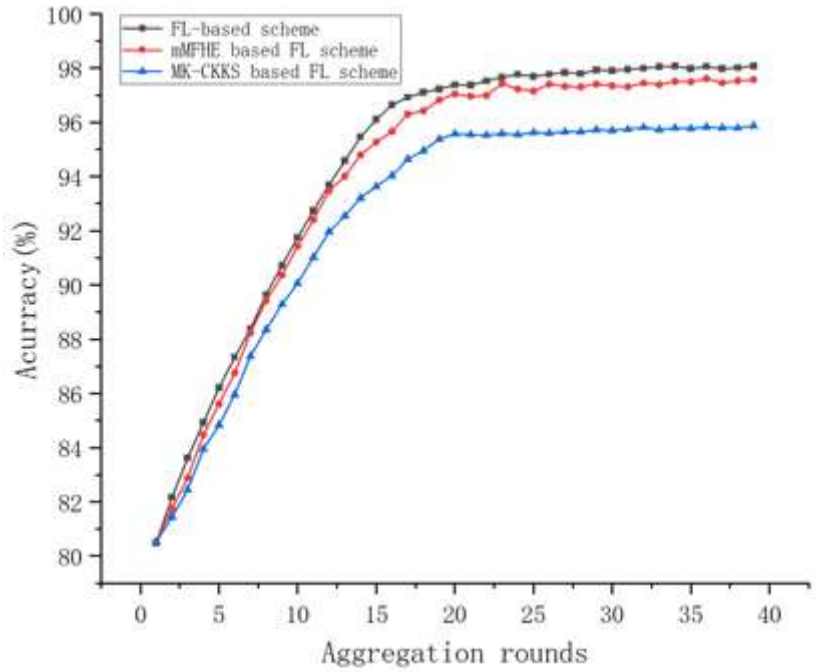

Figure 8. Model accuracy

7. Discussion

This section highlights key challenges associated with mMFHE-based federated learning and outlines potential future research directions.

One major concern is the **performance overhead associated with encrypted data transmission**. Although homomorphic encryption (HE) enables computations directly on encrypted data without requiring decryption—thus ensuring strong privacy protection—it introduces significant computational complexity. Compared to non-encrypted federated learning, HE-based systems incur additional latency due to encryption and decryption processes. Furthermore, advanced encryption techniques demand higher computational resources and memory, as also observed in the experimental evaluation of mMFHE. These factors can negatively impact system performance, particularly in resource-constrained environments or real-time applications, making mMFHE less suitable for latency-sensitive scenarios when compared to TEE-based approaches.

To mitigate these limitations, **reducing computational complexity** is an important research direction. Future work may explore hybrid cryptographic frameworks that combine homomorphic encryption with other techniques such as symmetric encryption or secure multi-party computation (SMPC). In such approaches, HE can be applied selectively to sensitive computations, while more efficient encryption methods handle less critical operations. Additionally, adaptive or selective encryption strategies can be developed to optimize encryption strength and granularity based on data sensitivity and computational requirements.

Another challenge lies in **key management** within multi-key fully homomorphic encryption systems. Managing multiple keys during multi-party decryption increases the risk of errors and system inefficiencies. To address this, future enhancements may incorporate a Trusted Execution Environment (TEE) as a Key Management Center (KMC), responsible for secure key generation, distribution, and lifecycle management. Appropriate key distribution protocols would also be necessary to ensure secure communication between the KMC and participating users.

Scalability is also a critical issue, particularly as the number of participating clients increases. The system may experience increased computational and

Doi:10.48047/ijiee.2026.16.2.43

communication burdens with large-scale participation. To overcome this, a dynamic participant management mechanism can be introduced, allowing efficient handling of user entry and exit during the training process. This mechanism should integrate secure registration, authentication, and key distribution for new participants, as well as proper revocation procedures when participants leave the system.

Finally, **interoperability** remains an important consideration in multi-organization federated learning environments. Ensuring seamless communication and secure data exchange across heterogeneous platforms requires a flexible and scalable system architecture. Future work should focus on designing modular frameworks that support plug-and-play integration of diverse computing nodes, storage systems, and third-party services while maintaining strict privacy guarantees.

8. Conclusions and Future Work

This study presents a privacy-preserving federated learning framework based on multi-key fully homomorphic encryption. Specifically, an enhanced MKFHE scheme, referred to as mMFHE, is developed using principles derived from the GSW13 scheme and secured under the Common Reference String (CRS) model. To address privacy leakage issues inherent in single-key FHE schemes within multi-user environments, the proposed approach incorporates key homomorphism, enabling the use of aggregated public keys and collaborative decryption.

Additionally, the mMFHE scheme supports packing multiple plaintext messages into a single ciphertext, improving computational efficiency while maintaining a low ciphertext expansion rate. Security analysis confirms that the proposed framework is robust against collusion attacks involving up to k=N−1k = N - 1k=N−1 participants and the central server. Experimental results further demonstrate its effectiveness in terms of computational performance, memory efficiency, and model accuracy.

Despite these advantages, certain challenges remain. One key limitation is the issue of **participant unavailability during training**. In multi-key environments, if a user becomes offline, their contribution to partial decryption is missing, potentially causing failure in the final decryption stage. This issue is particularly critical in large-scale and dynamic environments such as the Internet of Things (IoT).

Future research will focus on addressing these limitations by developing robust mechanisms to handle participant dropout, improving system scalability, and enhancing real-time applicability. These improvements will further strengthen the practicality of mMFHE-based federated learning in real-world distributed and privacy-sensitive applications.

References

1. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019) , Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
2. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep models under the GAN: Information leakage from collaborative deep learning. In
3. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 603–618.
4. Vatter, J.; Mayer, R.; Jacobsen, H.A. The evolution of distributed systems for graph neural networks and their origin in graph processing and deep learning: A survey. ACM Comput. Surv. 2023, 56, 1–37. [CrossRef]
5. McMahan, H.B.; Yu, F.; Richtarik, P.; Suresh, A.; Bacon, D. Federated learning: Strategies for improving communication efficiency.
6. In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 5–10.
7. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A.

Doi:10.48047/ijiee.2026.16.2.43

Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale,FL, USA, 20–22 April 2017; pp. 1273–1282.
8. Rieyan, S.A.; News, M.R.K.; Rahman, A.M.; Khan, S.A.; Zaarif, S.T.J.; Alam, M.G.R.; Hassan, M.M.; Ianni, M.; Fortino, G. An
9. advanced data fabric architecture leveraging homomorphic encryption and federated learning. Inf. Fusion 2024, 102, 102004. [CrossRef]
10. Mantey, E.A.; Zhou, C.; Anajemba, J.H.; Arthur, J.K.; Hamid, Y.; Chowhan, A.; Otuu, O.O. Federated learning approach for
11. secured medical recommendation in internet of medical things using homomorphic encryption. IEEE J. Biomed. Health Inform.2024, 28, 3329–3340. [CrossRef] [PubMed]
12. Hou, X.; Wang, J.; Jiang, C.; Meng, Z.; Chen, J.; Ren, Y. Efficient federated learning for metaverse via dynamic user selection, gradient quantization and resource allocation. IEEE J. Sel. Areas Commun. 2023, 42, 850–866. [CrossRef]
13. Ren, Y.; Lv, Z.; Xiong, N.N.; Wang, J. HCNCT: A cross-chain interaction scheme for the blockchain-based metaverse. ACM Trans. Multimed. Comput. Commun. Appl. 2024, 20, 1–23. [CrossRef]
14. Issa, W.; Moustafa, N.; Turnbull, B.; Sohrabi, N.; Tari, Z. Blockchain-based federated learning for securing internet of things: A comprehensive survey. ACM Comput. Surv. 2023, 55, 1–43. [CrossRef]
15. Aono,Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. 2017, 13, 1333–1345.
16. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706.
17. Sun, L.; Wang, Y.; Ren, Y.; Xia, F. Path signature-based xai-enabled network time series classification. Sci. China Inf. Sci. 2024,67, 170305. [CrossRef]
18. Ren, Y.; Zhu, F.; Wang, J.; Sharma, P.K.; Ghosh, U. Novel vote scheme for decision-making feedback based on blockchain in internet of vehicles. IEEE Trans. Intell. Transp. Syst. 2021, 23, 1639–1648. [CrossRef]
19. Mukherjee, P.; Wichs, D. Two round multiparty computation via multi-key FHE. In Advances in Cryptology—EUROCRYPT 2016,Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12May2016; Proceedings 31; Springer: Berlin/Heidelberg, Germany, 2012; pp. 735–763.
20. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. Proc.Mach. Learn. Syst. 2020, 2, 429–450.
21. Asad, M.; Moustafa, A.; Ito, T. Fedopt: Towards communication efficiency and privacy preservation in federated learning. Appl.Sci. 2020, 10, 2864. [CrossRef]
22. 18. Zhang, J.; Hua, Y.; Wang, H.; Song, T.; Xue, Z.; Ma, R.; Guan, H. Fedala: Adaptive local aggregation for personalized federatedlearning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37,pp. 11237–11244.
23. Yu,X.; Liu, R.; Nkenyereye, L.; Wang, Z.; Ren, Y. ACRS-Raft: A Raft Consensus Protocol for Adaptive Data Maintenance in the Metaverse Based On Cauchy Reed-Solomon Codes. IEEE Trans. Consum. Electron. 2024, 70, 3792–3801. [CrossRef]
24. hang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated
25. learning. In Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 20), Online, 15–17 July 2020;pp. 493–506.

Doi:10.48047/ijiee.2026.16.2.43

26. Madi, A.; Stan, O.; Mayoue, A.; Grivet-Sébert, A.; Gouy-Pailler, C.; Sirdey, R. A secure federated learning framework using homomorphic encryption and verifiable computing. In Proceedings of the 2021 Reconciling Data Analytics, Automation, Privacy and Security: A Big Data Challenge (RDAAPS), Hamilton, ON, Canada, 18–19 May 2021; pp. 1–8.
27. Stripelis, D.; Saleem, H.; Ghai, T.; Dhinagar, N.; Gupta, U.; Anastasiou, C.; Ver Steeg, G.; Ravi, S.; Naveed, M.; Thompson, P.M.; et al. Secure neuroimaging analysis using federated learning with homomorphic encryption. In Proceedings of the 17th International Symposium on Medical Information Processing and Analysis, Campinas, Brazil, 17–19 November 2021; Volume 12088, pp. 351–359.
28. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. AC Comput. Surv. 2018, 51, 1–35. [CrossRef]
29. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secureaggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and
30. Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
31. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.; Poor, H.V. Federated learning with differential privacy:Algorithms and performance analysis. IEEE Trans. Inf. Forensics Secur. 2020, 15, 3454–3469. [CrossRef]
32. Truex,S.; Liu, L.; Chow, K.H.; Gursoy, M.E.; Wei, W. LDP-Fed: Federated learning with local differential privacy. In Proceedings of the third ACM International Workshop on Edge Systems, Analytics and Networking, Heraklion, Greece, 27 April 2020; pp. 61–66.
33. Hu,R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized federated learning with differential privacy. IEEE Internet Things J. 2020,7, 9530–9539. [CrossRef]
34. Li, Y.; Zhou, Y.; Jolfaei, A.; Yu, D.; Xu, G.; Zheng, X. Privacy-preserving federated learning framework based on chained secure multiparty computing. IEEE Internet Things J. 2020, 8, 6178–6186. [CrossRef]
35. Gehlhar, T.; Marx, F.; Schneider, T.; Suresh, A.; Wehrle, T.; Yalame, H. SafeFL: MPC-friendly framework for private and robust federated learning. In Proceedings of the 2023 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 25 May 2023; pp. 69–76.
36. Zhang, J.; Chen, B.; Yu, S.; Deng, H. PEFL: A privacy-enhanced federated learning scheme for big data analytics. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
37. Li, Y.; Li, H.; Xu, G.; Huang, X.; Lu, R. Efficient privacy-preserving federated learning with unreliable users. IEEE Internet Things J. 2021, 9, 11590–11603. [CrossRef]
38. Ren, Y.; Leng, Y.; Qi, J.; Sharma, P.K.; Wang, J.; Almakhadmeh, Z.; Tolba, A. Multiple cloud storage mechanism based on blockchain in smart homes. Future Gener. Comput. Syst. 2021, 115, 304–313. [CrossRef]
39. He,C.; Liu, G.; Guo, S.; Yang, Y. Privacy-preserving and low-latency federated learning in edge computing. IEEE Internet Things J. 2022, 9, 20149–20159. [CrossRef]
40. Ren, Y.; Leng, Y.; Cheng, Y.; Wang, J. Secure data storage based on blockchain and coding in edge computing. Math. Biosci. Eng.2019, 16, 1874–1892. [CrossRef] [PubMed]
41. Cai, Y.; Ding, W.; Xiao, Y.; Yan, Z.; Liu, X.; Wan, Z. SecFed: A Secure and Efficient Federated Learning Based on Multi-Key Homomorphic Encryption. IEEE Trans. Dependable Secur. Comput. 2023, 21, 3817–3833. [CrossRef]
42. Ma,J.; Naas, S.A.; Sigg, S.; Lyu, X. Privacy-preserving federated learning

Doi:10.48047/ijiee.2026.16.2.43

based on multi-key homomorphic encryption. Int. J. Intell. Syst. 2022, 37, 5880–5901. [CrossRef]
43. Walskaar, I.; Tran, M.C.; Catak, F.O. A practical implementation of medical privacy-preserving federated learning using multi-key homomorphic encryption and flower framework. Cryptography 2023, 7, 48. [CrossRef]
44. Zhang, Q.; Jing, S.; Zhao, C.; Zhang, B.; Chen, Z. Efficient federated learning framework based on multi-key homomorphicencryption. In Advances on P2P, Parallel, Grid, Cloud and Internet Computing, Proceedings of the 16th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2021), Fukuoka, Japan, 28–30 October 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 88–105.
45. Wang, H.; Feng, Y.; Ding, Y.; Tang, S. A multi-key SMC protocol and multi-key FHE based on some-are-errorless LWE. Soft Comput. 2019, 23, 1735–1744. [CrossRef]
46. Gentry, C.; Sahai, A.; Waters, B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster,attribute-based. In Advances in Cryptology–CRYPTO 2013, Proceedings of the 33rd Annual Cryptology Conference, Santa Barbara, CA,USA, 18–22 August 2013; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2013; pp. 75–92.
47. Li, Z.; Ma, C.; Zhou, H. Multi-key FHE for multi-bit messages. Sci. China Inf. Sci. 2018, 61, 029101. [CrossRef]
48. Li, Z.; Ma, C.; Morais, E.; Du, G. Multi-bit Leveled Homomorphic Encryption via-Based. In Proceedings of the International
49. Conference on Information Security and Cryptology, Beijing, China, 4–6 November 2016; pp. 221–242.
50. Sun, L.; Li, C.; Ren, Y.; Zhang, Y. A Multitask Dynamic Graph Attention Autoencoder for Imbalanced Multilabel Time SeriesClassification. IEEE Trans. Neural Netw. Learn. Syst. 2024, 35, 11829–11842. [CrossRef]
51. Asharov, G.; Jain, A.; López-Alt, A.; Tromer, E.; Vaikuntanathan, V.; Wichs, D. Multiparty computation with low communication, computation and interaction via threshold FHE. In Advances in Cryptology–EUROCRYPT 2012, Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012; Proceedings 31; Springer: Berlin/Heidelberg, Germany, 2012; pp. 483–501.
52. López-Alt, A.; Tromer, E.; Vaikuntanathan, V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 19–2 May2012; pp. 1219–1234.

Doi:10.48047/ijiee.2026.16.2.43