Available Online: 19 March 2025 DOI: 10.54254/2977-3903/2025.21661

A secure aggregation method for federated learning based on homomorphic encryption

Xiaoge Ma

Shenzhen University, Shenzhen, China

1479966299@qq.com

Abstract. The development of cloud computing and big data has promoted the use of cloud servers in machine learning but has also raised concerns about privacy security. To enhance security and efficiency, this paper proposes a multi-key aggregation scheme based on improved Ring Learning With Errors (R-LWE) homomorphic encryption. This method protects the privacy of local model parameters and prevents information leakage through collaborative decryption. Experimental results demonstrate that the proposed scheme can resist collusion attacks, reduce communication overhead, and maintain model accuracy.

Keywords: federated learning, privacy protection, homomorphic encryption

1. Introduction

In the context of the integration of informatization and distributed computing [1], data resources have become a core element of modern industries, with cloud storage and collaborative modeling being widely applied. Traditional machine learning faces three major challenges [2]: limited cross-institutional data sharing, high risks associated with centralized data processing, and expensive cloud computing costs. In 2016, Google proposed federated learning [3], which avoids direct data transmission and instead exchanges only model parameters, achieving both privacy protection and cross-institutional collaboration. This study optimizes homomorphic encryption to improve computational efficiency and reduce communication complexity. The proposed scheme not only protects privacy but also significantly reduces communication overhead while achieving an accuracy level comparable to traditional federated learning on real-world datasets, demonstrating its practicality and efficiency.

2. Related work

The risk of private data leakage has become increasingly prominent, potentially leading to identity theft, financial fraud, and other security threats. Major global economies have established multi-level regulatory frameworks to address these challenges. In response, the academic community has proposed the integration of homomorphic encryption technology [4], which supports encrypted computations and prevents information leakage during gradient exchange. Federated learning has been applied in various fields, including computer vision, autonomous driving, and natural language processing [5], leading to the development of frameworks such as TensorFlow Federated and FATE [6]. However, it still faces challenges related to communication efficiency, resource imbalances, and vulnerability to malicious attacks. Existing privacy protection methods include differential privacy [7], fully homomorphic encryption [8], and secure multi-party computation [9]. For example, the DSGD algorithm reduces privacy leakage risks [10], Paillier encryption enhances model security [11], and secure aggregation protocols improve robustness [12]. Current research faces major challenges such as data heterogeneity and device resource disparities. Low-quality data may degrade global model accuracy, while existing differential privacy mechanisms have security limitations in non-identity (non-ID) scenarios [13]. Moreover, significant computational power differences among edge devices can affect training efficiency. The FedCS protocol [14] optimizes training through dynamic node selection, but further improvements in resource scheduling strategies are still needed [15]. As an important approach in privacy computing, homomorphic encryption enables secure encrypted computations, enhancing data security. This study focuses on optimizing privacy protection in federated learning through homomorphic encryption, providing technical support for the development of privacy-preserving computing infrastructures.

3. System model and algorithm

To address privacy leakage issues, this paper proposes an improved privacy-preserving federated learning scheme with multi-key aggregation (PFLMA). This scheme optimizes Ring Learning With Errors (R-LWE) homomorphic encryption, improving computational efficiency. Based on the MK-CKKS scheme, it innovatively employs aggregated public key encryption to secure local model parameters. As a result, the cloud server can only decrypt the aggregated result without accessing individual model updates, effectively protecting privacy. Additionally, this scheme resists collusion attacks between participants and the cloud server.

3.1. System model

The PFLMA scheme involves three main entities: the Key Generation Center (KGC), the Cloud Server (CS), and the Participants (P). The system workflow consists of key generation, model training, encrypted transmission, aggregation, and update processes (see Figure 1). The KGC generates public-private key pairs for participants and provides public parameters. Each participant trains a local model and uploads encrypted parameters to the CS. The CS aggregates the encrypted data and distributes the global model. The participants then continue training until convergence.

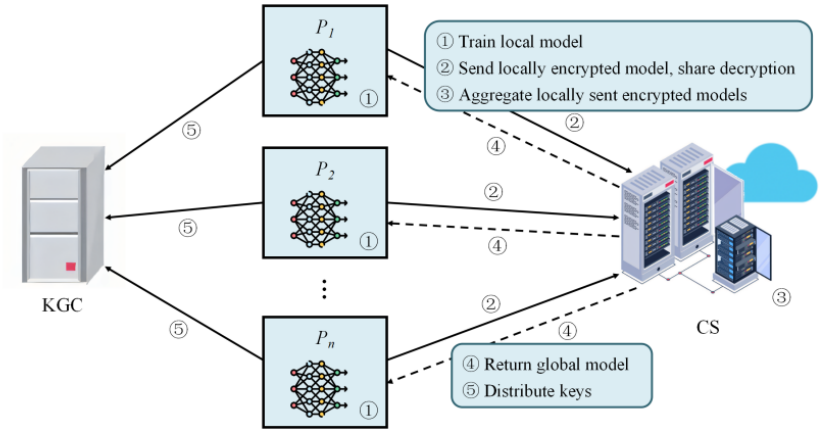

Figure 1. Illustrates the system model of the proposed PFLMA scheme

Key Generation Center (KGC): A trusted third party responsible for generating public parameters, managing key lifecycles, and coordinating secure interactions. The KGC ensures parameter security and supports key updates and revocation.

Participants (P): Data providers that engage in local model training. They encrypt their model parameters using the keys generated by the KGC before uploading them to the CS. The participants optimize their local models while preserving privacy.

Cloud Server (CS): Responsible for coordinating and aggregating encrypted model parameters to generate a global model and distributing it back to participants. The CS does not have direct access to raw data and can only recover encrypted aggregate results.

3.2. Scheme implementation

The PFLMA scheme employs multi-key homomorphic encryption, allowing multiple participants to independently encrypt data and perform collaborative computation, ensuring the secure transmission of model parameters. The scheme improves R-LWE homomorphic encryption, utilizes Stochastic Gradient Descent (SGD) for model training, and incorporates a key management mechanism to ensure privacy security.

3.2.1. System initialization phase

During this phase, the KGC and participants complete system initialization through the following steps:

(1) Initialization: The KGC selects a security parameter, sets $\theta = 2^k, k \geq 1$, and confirms the key distribution χ and error distribution ψ, with ciphertext modulus q.

The KGC sends public parameters $pp = (\theta, q, \chi, \psi, a)$ to the participants.

(2) Encoding and Decoding: Participants expand model parameters into vectors, normalize them, and encode them as polynomials in ring R.

(3) Key Generation: Each participant P_i generates a private key $sk_i = s_i \leftarrow \chi$ and an error vector $e_i \leftarrow \psi$, then computes the public key $pk_i = b_i = -s_i \cdot a + e_i \, mod \, q$. The KGC aggregates the keys to generate a global public key $pk = \sum_{i=1}^{n} b_i$ and an aggregated private key $sk = \sum_{i=1}^{n} sk_i$.

3.2.2. Local training phase

(1) Encryption: Participants use SGD to train local models, obtaining model weights w_t^i.

The public key $\tilde{b}$ encrypts plaintext m_i into ciphertext $cm_i = (c_{i0}, c_{i1})$, incorporating randomness $v \leftarrow \chi$. The encryption formula is as follows:

$$cm_i = (c_{i0} + c_{i1}) = (v \cdot \tilde{b} + e_{i0} + m_i, v_i \cdot a + e_{i1}). \quad (1)$$

(2) Key Switching: Participants compute a transformation key $R_i = (-sk_i \cdot a + sk \cdot a + se_i) \, mod \, q$ and perform ciphertext conversion using R_i. The transformed ciphertext cm_i' is then sent to the cloud server.

(3) Homomorphic Addition: The cloud server aggregates all participants' ciphertexts using homomorphic addition, yielding the aggregated ciphertext $C_{sum} = \sum_{i=1}^{n} cm_i'$, where

$C_{sum0} = \sum_{i=1}^{n} c_{i0}'$ and $C_{sum1} = \sum_{i=1}^{n} c_{i1}'$.

3.2.3. Partial decryption phase

Each participant partially decrypts the aggregated ciphertext using their private key s_i, and each participant decrypts the aggregated ciphertext sent by the cloud server, generating the partial decryption result $D_i = C_{sum1} - s_i \cdot C_{sum0} \, mod \, q$.

3.2.4. Aggregation phase

The cloud server aggregates the partial decryption results from all participants to recover the plaintext and compute the sum of parameters:

$$\widetilde{m} \approx \sum_{i=1}^{n} D_i \, mod \, q \, . \quad (2)$$

3.2.5. Model update phase

The cloud server and participants collaboratively update the local training models to derive the final global model. The cloud server computes the weighted average of all participants' local models to obtain the global model w_{t+1}:

$$w_{t+1} = \frac{1}{n} \sum_{i=1}^{n} w_t^i. \quad (3)$$

The cloud server then transmits the updated global model parameters to each participant for the next training round until convergence.

3.3. Security analysis

The PFLMA scheme ensures the confidentiality of model parameters in federated learning, preventing data leakage. The security analysis focuses on the cloud server, participants, and collusion resistance.

(1) Security Against an Honest-but-Curious Cloud Server: The cloud server can only process encrypted data and cannot access any plaintext information. Even after aggregating encrypted data, the server can only obtain the sum, not individual participants' models or training data.

(2) Security Against Honest-but-Curious Participants: Each participant encrypts its own data, preventing them from inferring others' private information. The encryption randomness and error terms ensure that even if one participant decrypts their data, they cannot obtain others' information.

(3) Security Against Collusion Between Participants and the Cloud Server: Even if the cloud server and some participants collude, they cannot obtain other participants' private data. The encryption and randomness mechanisms prevent effective decryption of ciphertext, ensuring data privacy.

4. Experimental analysis

4.1. Experimental setup

The experiments were conducted on a Windows 10 platform, with a hardware configuration consisting of an Intel i5-1035G1 processor and 16GB RAM. The PFLMA scheme was simulated in the context of horizontal federated learning. The experiments

utilized two publicly available datasets from UCI Machine Learning Repository: Heart Disease and Pima. The study compared three federated learning approaches:

PFLMA (Proposed Scheme)

Federated Learning Based on MK-CKKS

Unencrypted Federated Learning (UFL)

The evaluation focused on model accuracy and communication overhead.

4.2. Accuracy analysis

The training time of the PFLMA scheme increases as the number of participants grows. Experiments were conducted with different numbers of local training rounds N, specifically 1, 5, 10, 15, 20, 30, and 40 rounds. The results showed that (see Figure 2 and 3):

When N=20, the accuracy of PFLMA on the Heart Disease and Pima datasets was 92.73% and 93.37%, respectively. These values were very close to UFL (93.25% and 93.90%), and significantly better than the MK-CKKS scheme.

When N=40, the accuracy of PFLMA reached 92.89% and 93.87%, which was almost on par with UFL (93.65% and 94.53%) and exceeded the MK-CKKS scheme by 1.25% and 1.89%, respectively.

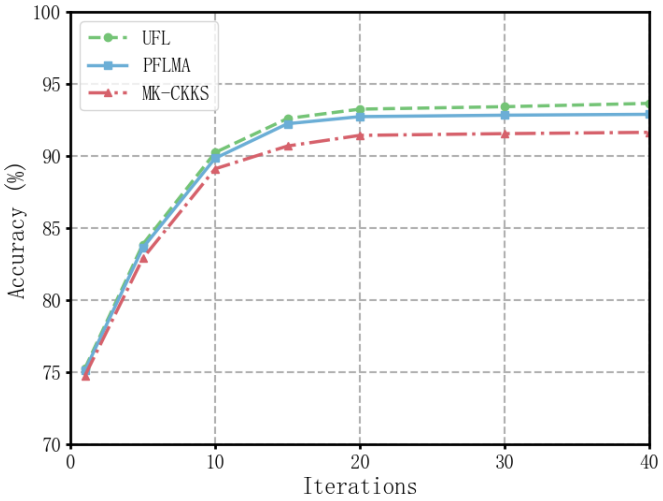

Figure 2. Accuracy on heart disease dataset

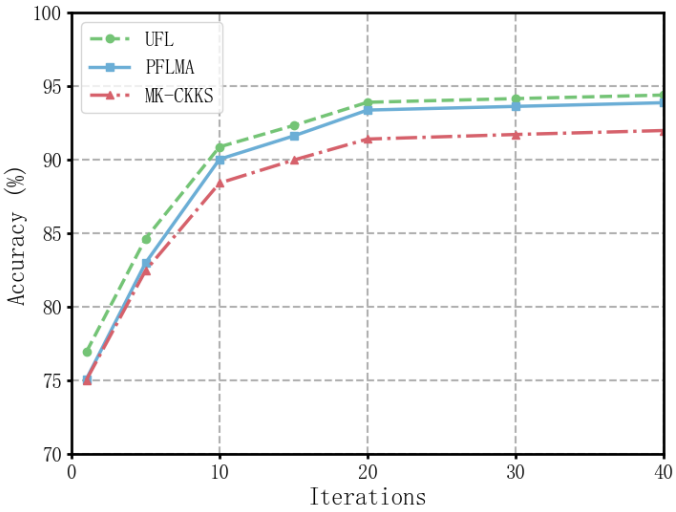

Figure 3. Accuracy on Pima dataset

The results indicate that as the number of training rounds increases, PFLMA achieves accuracy close to UFL while outperforming the MK-CKKS scheme.

4.3. Communication efficiency analysis

Each model consists of 510 weights, with each weight occupying 64 bits, resulting in a ciphertext size of approximately 92KB. The PFLMA scheme introduces additional collaborative decryptions C_{sum1} and D_i of 52KB. To reduce communication overhead, experiments were conducted with 10 aggregation rounds or 5 aggregation rounds, allowing convergence to be achieved with 20 or 40 local training rounds, respectively (see Figure 4).

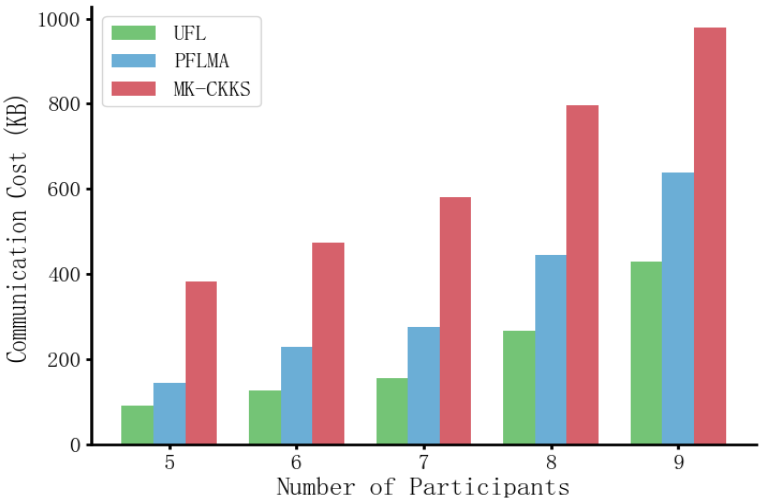

Figure 4. Communication efficiency comparison

The results demonstrate that the communication overhead of PFLMA is slightly higher than UFL but significantly lower than the MK-CKKS scheme. By optimizing computational structures and introducing a trusted key management center, PFLMA reduces both computational complexity and communication costs.

5. Conclusion

This paper proposes an efficient privacy-preserving federated learning scheme based on multi-key aggregation (PFLMA). The scheme improves upon traditional R-LWE homomorphic encryption by simplifying related computations, enhancing computational efficiency and decryption accuracy, and reducing storage complexity during computation and communication, thereby improving overall efficiency. Furthermore, by refining the MK-CKKS scheme, the proposed method defines an aggregated public key and shared decryption mechanism, ensuring parameter privacy during federated learning model updates and effectively preventing collusion attacks between participants and the cloud server. The introduction of a trusted key generation center also minimizes communication overhead among participants. Both theoretical analysis and experimental results confirm that PFLMA effectively preserves data privacy, significantly reduces communication costs, improves efficiency, and performs competitively on real-world datasets.

References

[1] Sha, J., Ebadi, A. G., Mavaluru, D., Alshehri, M., Alfarraj, O., & Rajabion, L. (2020). A Method for Virtual Machine Migration in Cloud Computing Using a Collective Behavior-based Metaheuristics Algorithm. *Concurrency and Computation: Practice and Experience, 32*(2), e5441.

[2] Wei, L. F., Chen, C. C., Zhang, L., Li, M. S., Chen, Y. J., & Wang, Q. (2020). Security Issues and Privacy Protection in Machine Learning. *Journal of Computer Research and Development, 57*(10), 2066-2085.

[3] McMahan, H. B., Moore, E., Ramage, D., & Agüera y Arcas, B. (2016). Federated Learning of Deep Networks Using Model Averaging. *arXiv preprint arXiv:1602.05629, 2*(2), 15-18.

[4] Yang, Y. T., Zhao, Y., Zhang, J. M., Huang, J. R., & Gao, Y. (2021). Progress in Homomorphic Cryptography Theory and Applications. *Journal of Electronics & Information Technology, 43*(02), 475-487.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

[6] Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., & Passerat-Palmbach, J. (2018). A Generic Framework for Privacy-preserving Deep Learning. *arXiv preprint arXiv:1811.04017*.

[7] Shokri, R., Shmatikov, V. (2015). Privacy-preserving Deep Learning. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1310-1321.

[8] Phong, L. T., Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2017). Privacy-preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security, 13*(5), 1333-1345.
[9] Jayaraman, B., & Wang, L. (2018). Distributed Learning without Distress: Privacy-preserving Empirical Risk Minimization. *Advances in Neural Information Processing Systems, 31*.
[10] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308-318.
[11] Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., & Thorne, B. (2017). Private Federated Learning on Vertically Partitioned Data via Entity Resolution and Additively Homomorphic Encryption. *arXiv preprint arXiv:1711.10677*.
[12] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2016). Practical Secure Aggregation for Federated Learning on User-held Data. *arXiv preprint arXiv:1611.04482*.
[13] Zhao, L., Wang, Q., Zou, Q., Zhang, Y., & Chen, Y. (2018). Privacy-preserving Collaborative Deep Learning with Irregular Participants. *arXiv preprint arXiv:1812.10113*.
[14] Nishio, T., Yonetani, R. (2019). Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. *ICC 2019 - IEEE International Conference on Communications (ICC)*, 1-7.
[15] Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K., & Yonetani, R. (2020). Hybrid-FL for Wireless Networks: Cooperative Learning Mechanism using non-IID Data. *ICC 2020 - IEEE International Conference on Communications (ICC)*, 1-7.