Enhancing Privacy and Efficiency in Federated Learning Through Hybrid Homomorphic Encryption

Rahool Dembani[1,2,3](✉), Ioannis Karvelas[1], Stamatia Rizou[1], and Domenico Tegolo[3]

[1] R&D and Innovation Department, SingularLogic, Athens, Greece
rdembani@singularlogic.eu, rahool.dembani@studenti.unime.it
[2] Department of Mathematics and Computational Sciences, University of Messina, Messina, Sicily, Italy
[3] Department of Mathematics and Computer Science, University of Palermo, Palermo, Italy

Abstract. Federated Learning (FL) allows the training of models over distributed data sources without compromising the privacy of users in different client devices. Nonetheless, encryption mechanisms, including Homomorphic Encryption (HE) and symmetric encryption, such as the Advanced Encryption Standard (AES), enhance security but usually come at the cost of computational expense, affecting the speed and scale of the model. To tackle these issues, we propose a hybrid HE model which integrates the CKKS method of HE with AES encryption to ensure privacy and computational performance. Our experiments with an agricultural crop production dataset indicate that the proposed model is considerably more efficient regarding predictive performance and training time than the normal encryption model. The hybrid method also outperforms security and performance resource-constrained systems compared to systems that use only AES or HE. This implies that a hybrid strategy can be used in FL with the objective of achieving both security and efficiency.

Keywords: Federated learning · Homomorphic encryption · Hybrid encryption · CKKS scheme · Privacy preservation

1 Introduction

The growing spread of data generated by edge devices and the Internet of Things (IoTs) has necessitated the creation of distributed machine-learning methods. In this context, Konečný et al. (2017) played a significant role in promoting Federated Learning (FL) as a new concept which makes it possible for various clients to participate in the construction of a network-based model without revealing their original data [1], hence ensuring that the privacy is respected and legal norms, such as the General Data Protection Regulation (GDPR) [2], are followed. FL has been reported across numerous sectors, including healthcare [3], finance [4], and IoT [5]. The standard FL workflow consists of the individual model being trained locally, sending the model updates to a central server and subsequently collating these updates so that the global model can be enhanced. FL, despite

K. Arai (Ed.): IntelliSys 2025, LNNS 1567, pp. 689–696, 2025.
https://doi.org/10.1007/978-3-032-00071-2_42

its advantages, encapsulates privacy infringements issues. Model inversion attack makes it possible to extract some private information which is transmitted during the updates of the model from the clients to a central server [6]. As a counter-measure against such risks, the FL architecture demands secure encryption mechanisms. First, Gentry (2009) proposed the concept of fully homomorphic encryption (FHE) using ideal lattices. Later, the CKKS scheme was developed for approximate arithmetic on encrypted real numbers [7]. It is intended to perform computations on encrypted data, avoiding decryption during both the training and the aggregation operations. So, of course, HE is computationally expensive. This entails an increased training time as well as a higher consumption of resources, which causes problems for resource-limited types of applications, for example, IoT. On the other hand, as noted by Daemen and Rijmen (2002), encryption schemes like the Advanced Encryption Standard (AES) facilitate low overhead secure communication of data [8] however, they are unable to perform computations on ciphertexts. This limitation entails the frequent decryption of model parameters prior to their aggregation to resolve privacy concerns and utilize FL's advantages further. To address these limitations, we proposed a Hybrid Homomorphic Encryption model that involved integrating the CKKS approach of HE and the AES encryption in the ecosystem of FL. This hybrid approach applies AES to ensure the safe transfer of information. As a result of it, significant degradation of the CKKS enables secure merging of all the changes in the model without decrypting any information. The suggested model thus preserves privacy whilst enhancing computational efficiency by combining these two encryptions.

The combination of these two approaches allows for lowering the computational complexity encumbered by the full-scale homomorphic encryption. While AES is useful and effective encryption for cipher and time communication, CKKS is effective as it allows computation on an encrypted input with no decryption needed at all, which is crucial in minimizing privacy concerns during the model update aggregation.

The remainder of this paper is organized as follows. In Sect. 2, we provide an overview of privacy-preserving encryption techniques, including HE, symmetric encryption (AES), and Paillier encryption. Section 3 introduces the proposed hybrid homomorphic encryption model, detailing its mathematical formulation and integration into the FL framework. Section 4 describes the experimental setup, including the dataset, model architecture, and evaluation metrics. Section 5 presents the results of our experiments, comparing the performance of the proposed hybrid model with other encryption methods. Finally, Sect. 6 discusses the implications of our findings, outlines future research directions, and concludes the paper.

2 Privacy-Preserving Encryption Techniques

The privacy of users takes precedence during the training of any machine learning models regarding FL model update aggregations. Several solutions have been created to protect the processes of FL from breaches.

2.1 Homomorphic Encryption

Homomorphic encryption supports arithmetic operations on ciphertexts without the need for decryption [7]. Likewise, data can be processed without the need to decrypt it in the CKKS system [9].

2.2 Symmetric Encryption (AES)

AES [8] is a symmetric cryptographic algorithm which performs well in terms of time efficiency. Even though it provides data transfer confidentiality protection, encrypted data computations are not possible, thus constraining scenarios where encrypted computations can be performed.

2.3 Paillier Encryption

Because of its potential to enable the homomorphic computation of ciphertexts, the encryption scheme known as Paillier was proposed by Paillier in 1999. Paillier encryption scheme is particularly notable as one of the probabilistic public-key cryptosystems. The exceptional part of this system is its potential to perform computations on the ciphertext without revealing the confidentiality level of such computations [10].

Considering these limitations, it becomes obvious that there is a need for a mixture model that takes advantage of AES's computational efficiency and the privacy afforded by the homomorphic encrypted model. HE is our proposed Hybrid Encryption model that will help eliminate this challenge by integrating the two encryption techniques to create a safe and productive FL model.

3 Proposed Methodology

The purpose of the proposed Hybrid Homomorphic Encryption model is to increase the confidentiality of transmitted and aggregated model updates while decreasing the computation burden by integrating the HE CKKS algorithm and AES encryption inside the FL framework.

Mathematically, the encryption process for each client's local model parameters $\theta_i^{(t)}$ is defined as:

$$\text{Hybrid\_ Enc}\left(\theta_i^{(t)}\right) = \text{AES\_ Enc}\left(\text{CKKS\_ Enc}\left(\theta_i^{(t)}\right)\right) \tag{1}$$

Here, CKKS_ Enc encrypts the model parameters using the CKKS scheme, enabling homomorphic operations while AES_ Enc further encrypts the CKKS ciphertext to secure the data during transmission.

Upon receiving the encrypted updates, the server performs homomorphic aggregation without decrypting individual updates:

$$\text{Enc}_{\text{CKKS}}\left(\theta_{\text{global}}^{(t+1)}\right) = \frac{1}{N}\sum\nolimits_{i=1}^{N}\text{CKKS\_ Enc}\left(\theta_i^{(t)}\right) \tag{2}$$

Finally, the aggregated encrypted global model is decrypted to update the global parameters:

$$\theta_{\text{global}}^{(t+1)} = \text{CKKS\_ Dec}\left(\text{Enc}_{\text{CKKS}}\left(\theta_{\text{global}}^{(t+1)}\right)\right) \tag{3}$$

This hybrid framework is rather unique in making sure the private updates of the individual models do not get transmitted or aggregated, therefore ensuring that the global model can be formulated even without the server having direct access to the raw data.

In the proposed hybrid approach, privacy has been preserved during two critical phases of the FL process: model updates and aggregation. We, however, in contrast to conventional approaches, by combining the two techniques of data encryption for secure transfer and key-locked CKKS for secure aggregation, allow the server never to see any raw data even when aggregates are created thus reducing the threat posed by model inversion and inference attacks. Examples of such methods are AES based methods that allow for quick transfer of model updates and shield them from prying ears, while CKKS permits model parameterized global models to be built without revealing the underlying models.

Figure 1 depicts client-side operations beginning with each client training a local model. Once trained, local model updates are encrypted using the CKKS (HE) technique, with the encryption context properly configured to strike a balance between security and computational performance. The homomorphically encrypted model updates are then serialized and further secured by encrypting the serialized data with AES and a symmetric key, yielding doubly encrypted model updates. These securely encrypted changes are then transferred to the server. On the server side, the operation begins by decrypting the received data with the symmetric AES key to obtain the homomorphically encrypted model updates. The server then securely aggregates these encrypted updates using homomorphic addition and scalar multiplication. Following aggregation, the combined model updates are decrypted using the CKKS decryption method. Finally, the decrypted aggregated updates are used to update the global model, ensuring that the entire system remains secure and efficient throughout the training process.

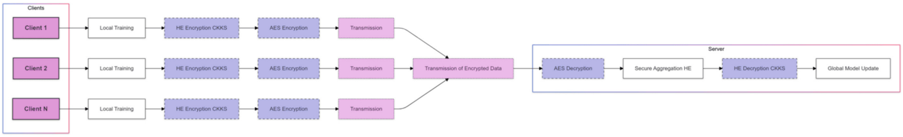

Fig. 1. Proposed hybrid homomorphic encryption model

The model ensures data privacy at two levels through advanced encryption techniques. First, it employs HE to secure model changes during the aggregation process, effectively preventing the server from accessing individual updates. This ensures that sensitive information remains protected while computations are performed. Additionally, the model incorporates AES encryption to safeguard the homomorphically encrypted data during transmission. This adds an extra layer of security, shielding the data from potential network threats and ensuring comprehensive privacy protection throughout the entire process.

The hybrid approach addresses possible weaknesses associated with either strategy when employed in isolation. Specifically, AES provides fast encryption for data transmission, whereas HE assures that even if the transmission is intercepted, the model updates remain safe during aggregation.

4 Experimental Setup

The experiment was performed using the machine [Intel® Core™ i7-5600U CPU @2.60 GHz and 8.00 GB RAM]. We used Kaggle's Agriculture Crop Yield dataset [11], picking 20,000 rows out of 1,000,000 due to limited computational resources and the inclusion of ten characteristics, including categorical (Region, Soil Type, Crop Type, Fertilizer, Irrigation, Weather) and numerical variables (Rainfall, Temperature, Days to Harvest). We preprocessed the data by removing rows with missing values, label encoding categorical features, normalizing numerical data with StandardScaler, and dividing it into 80% training and 20% testing. This preparation makes the dataset acceptable for testing the scalability and efficacy of the proposed encryption techniques. Our neural network model has a simple architecture that focuses on the effects of various encryption methods on FL performance. The network consists of nine input nodes, two hidden layers with 16 and 8 neurons that use ReLU activations, and a single output neuron for crop yield prediction. This simple structure assures that any performance discrepancies are due to encryption methods rather than model complexity, allowing for a transparent examination of how encryption affects the effectiveness and efficiency of FL on large, heterogeneous agricultural datasets.

Five different models were considered, namely: FL Baseline, which does not use encryption, FL with AES encryption, FL with CKKS homomorphic encryption FL with Paillier encryption and FL with HE (CKKS) + AES, which is the proposed model. In all the scenarios, 15 rounds of federated learning with 6 clients were conducted to mimic the realistic distribution setting.

5 Results and Discussions

The performance metrics include Mean Squared Error (MSE) Loss, Root Mean Squared Error (RMSE), R-squared R^2, and Training Time. These are mathematically defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2 \tag{4}$$

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{5}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2} \tag{6}$$

where y_i are the true values, $\widehat{y_i}$ are, the predicted values and $\overline{y}$ is the mean of the observed data.

Table 1. Performance metrics across FL methods

Model	MSE Loss	RMSE	R-squared (R^2)	Training Time (s)
FL with No Encryption (Baseline)	0.2624	0.5123	0.9100	16.49
FL with AES Encryption	0.2578	0.5077	0.9115	15.53
FL with Homomorphic Encryption (CKKS)	0.2564	0.5063	0.9120	23.92
FL with Paillier Encryption	0.2591	0.5090	0.9111	1167.47
FL with Hybrid Encryption (HE (CKKS) + AES) (Our Model)	**0.2536**	**0.5035**	0.9130	34.94

Table 1 and Fig. 2 show comparative studies of different FL models, highlighting variances in performance and computing efficiency. FL with Hybrid Encryption (HE(CKKS) + AES) achieved the highest accuracy, with the lowest MSE (0.2536) and RMSE (0.5035), as well as the balanced R-squared value (0.9130). The training time was 34.94 s. In contrast, FL with Homomorphic Encryption (CKKS) had a slightly higher MSE (0.2564) and RMSE (0.5063) but a strong R-squared (0.9120) with a training time of 23.92 s. The model that simply used AES encryption worked well (MSE: 0.2578, RMSE: 0.5077, R-squared: 0.9115) and took the least amount of time (15.53 s). The Paillier encrypted model performed comparably (MSE: 0.2591, RMSE: 0.5090, R-squared: 0.9111), but it took substantially longer to train (1167.47 s), making it the least efficient. The unencrypted model had an MSE of 0.2624, an RMSE of 0.5123, and an R-squared value of 0.9100 with a training time of 16.49 s, suggesting that encryption approaches can enhance model accuracy without significantly increasing training times.

The hybrid model strikes a balance between security and efficiency, making it suitable for environments with limited computational resources.

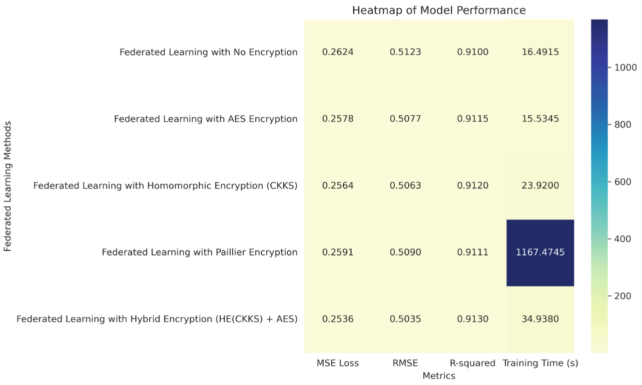

Fig. 2. Performance comparison of different encryption methods used in federated learning

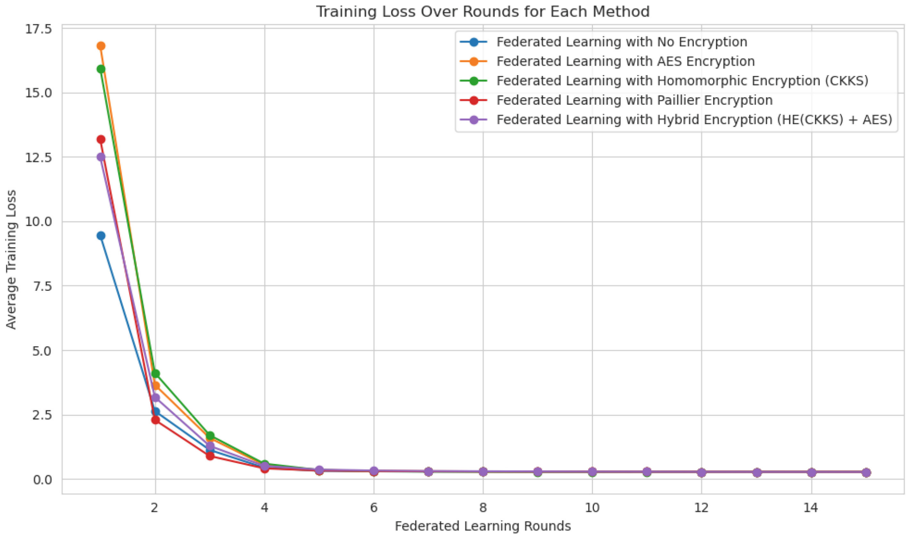

Fig. 3. Average training loss over FL rounds for each method

Figure 3 shows that the proposed model converges more rapidly and achieves lower training loss than other methods. Despite the added encryption layers, this indicates effective learning and stability in the training process.

The results obtained illustrate the practical advantages of our hybrid encryption approach (CKKS + AES). The hybrid method effectively combines AES's low computational overhead for secure data transmission with CKKS's capability to securely aggregate encrypted data, thus achieving higher accuracy and efficiency than standalone encryption schemes. The observed improvement in predictive performance and manageable training time demonstrates the proposed model's suitability for real-world FL applications, especially in resource-constrained environments like IoT and edge computing scenarios.

6 Conclusion

This work describes a Hybrid HE model that effectively balances privacy protection and computational performance in FL. The model protects model changes during transmission and aggregation by integrating the CKKS homomorphic encryption technique with AES encryption. Compared to typical encryption methods, experimental results on a large-scale agricultural dataset show improved prediction performance and shorter training times. Future work will include scalability testing to assess the model's performance with a larger number of clients and under different network conditions, extending the hybrid encryption approach to more complex machine learning models such as deep neural networks, experimenting with dynamic encryption parameters based on computational resources and security requirements, and investigating the model's resilience to advanced privacy attacks such as inference and reconstruction attacks. Future research can improve the security and efficiency of FL systems by refining and expanding on this hybrid method, paving the door for their use in a wider range of applications.

Acknowledgments. This work was funded by the European Commission under the Doctoral Networks Programme (MSCA-DN-101073381–EnTrust) within the Horizon Europe (HORIZON) Marie Skłodowska-Curie Actions.

References

1. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P, Suresh, A.T., Bacon, D.: Federated Learning: Strategies for Improving Communication Efficiency, Oct. 30 (2017). arXiv. https://doi.org/10.48550/arXiv.1610.05492
2. P. Voigt and A. Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide (2017). https://doi.org/10.1007/978-3-319-57959-7
3. Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S. : Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2018. LNCS, vol. 11383. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11723-8_9
4. Yang, Q, Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2), 12:1–12:19 (2019). https://doi.org/10.1145/3298981
5. Lim, W.Y.B., et al.: Federated learning in mobile edge networks: a comprehensive survey. IEEE Commun. Surv. Tutorials **22**(3), 2031–2063 (2020). https://doi.org/10.1109/COMST.2020.2986024
6. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, in CCS 2017, pp. 603–618. Association for Computing Machinery, New York (Oct 2017). https://doi.org/10.1145/3133956.3134012
7. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on Theory of computing, in STOC 2009, pp. 169–178. Association for Computing Machinery, New York (May 2009). https://doi.org/10.1145/1536414.1536440
8. Daemen, J., Rijmen, V:. The Advanced Encryption Standard Process. In: The Design of Rijndael: AES — The Advanced Encryption Standard, pp. 1–8. Springer, Berlin (2002). https://doi.org/10.1007/978-3-662-04722-4_1
9. Lee, Y., Lee, J.-W., Kim, Y.-S., No, J.-S.: Near-optimal polynomial for modulus reduction using L2-norm for approximate homomorphic encryption. IEEE Access **8**, 144321–144330 (2020). https://doi.org/10.1109/ACCESS.2020.3014369
10. Sakurai, K., Takagi, T.: New Semantically Secure Public-Key Cryptosystems from the RSA-Primitive. In: Naccache, D., Paillier, P. (eds.) Public Key Cryptography, pp. 1–16. Springer, Berlin (2002). https://doi.org/10.1007/3-540-45664-3_1
11. Agriculture Crop Yield Kaggle dataset https://www.kaggle.com/datasets/samuelotiattakorah/agriculture-crop-yield