ISSN: 2664-9640

# Enhanced Privacy-Preserving Federated Learning with Exit Tolerance and Verifiable Aggregation

Yudong Lin*

School of Electronic and Information Engineering, Wuyi University, Jiangmen 529020, China

*Corresponding Author: linyudong22@126.com

## Abstract

**With the escalating importance of data privacy, Federated Learning (FL) has drawn significant attention as a machine learning framework that facilitates collaborative model training among multiple users without the need for direct data sharing. This paper presents an enhanced Federated Learning scheme designed to bolster user privacy protection while maintaining the efficiency and performance of model training. Our approach integrates the use of random seeds and One-Time Password (OTP) technology to reinforce data encryption, and employs an advanced masking mechanism coupled with bilinear pairing for verification, thereby enhancing the security of the aggregation process. Additionally, our design accommodates user exits during the training process without compromising the overall training outcome. Through rigorous experimental analysis, we have demonstrated the effectiveness of our scheme, showcasing its acceptable computational and communication overheads. This research introduces a novel solution to privacy preservation challenges in Federated Learning, laying a solid foundation for the advancement of related technologies.**

## Keywords

**Federated Learning; Privacy Protection; Exit Tolerance; Masking; Bilinear Pairing.**

## 1. Introduction

The concept of "Federated Learning" was initially proposed by McMahan et al.[1], describing a mechanism that enables machine learning model training using data from various sources without the need for centralized data storage. Within the Federated Learning (FL) framework, multiple clients interact with a central server over multiple rounds to train a model. Each data source has a FL client deployed. Initially, the FL server distributes the same initial model to all clients. During each round of the FL process, clients train on their local data to refine the machine learning model they received and send these updates back to the server. The server then aggregates these models by calculating their average and sends the aggregated model back to the clients. After receiving the updated model, clients initiate a new round of the FL process, and this cycle continues between the clients and the server until the model converges, marking the end of the FL process. The core objective of FL is to protect the privacy of data while leveraging it to train a shared model.

Federated Learning facilitates the collaboration of multiple users with a single server to train a high-performance model. Users can complete model training by sharing their gradients without the need to share local data; however, studies have shown that shared gradients still retain privacy data used for training. In the Federated Learning framework, the server is typically an untrusted entity, meaning it may attempt to infer user privacy data from shared gradients, leading to severe privacy breaches. Moreover, the server may also forge aggregation results for its own benefit, greatly affecting the efficiency and performance of Federated Learning model training.

ISSN: 2664-9640

Currently, a multitude of solutions have been studied for the secure aggregation of gradients in Federated Learning. Among them, privacy protection schemes based on dual masking encryption introduce additional communication rounds in cases of user disconnection, while those based on homomorphic encryption employ heavy cryptographic computations, resulting in computational load[2].However, with technological advancements, enhancing privacy protection and security performance in Federated Learning has become a hot topic and challenge. This paper addresses the privacy leakage risks and security issues present in existing Federated Learning schemes by proposing a verifiable privacy-preserving Federated Learning scheme with exit tolerance capabilities. Our design enhances the security of data during transmission and processing by introducing random seeds and One-Time Password (OTP) technology. Concurrently, it ensures the correctness and security of model aggregation through an improved masking mechanism and bilinear pairing verification methods. Additionally, our scheme supports users exiting the training process at will without impacting other users, enhancing the system's robustness and practicality. The research outcomes of this paper not only provide new perspectives and solutions to privacy protection issues in Federated Learning but also offer theoretical support and practical guidance for the development and application of related technologies.

## 2. System Model

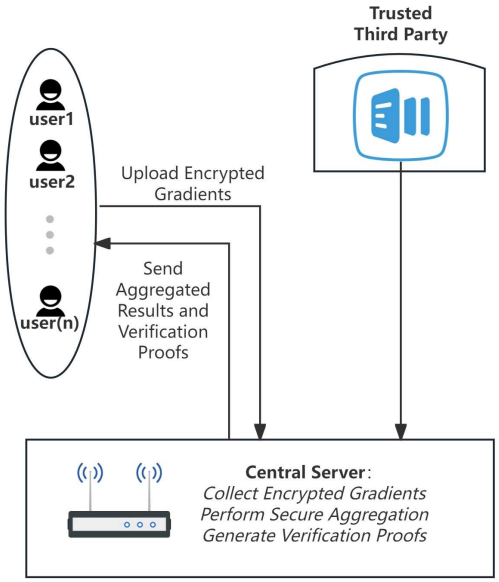


**Fig. 1** System Model

In this study, we have meticulously crafted an innovative Federated Learning (FL) scheme, designed to enhance data privacy protection through an integrated system model, while simultaneously improving the efficiency and performance of model training. As depicted in Fig. 1, this system model comprises three core components: a trusted third party, a central server, and a widespread array of user terminals. This design ingeniously balances the privacy needs of user data with the centralized requirements of model training.

The trusted third party plays a crucial role in the system, being responsible for generating global parameters and keys during the system initialization phase. These keys are the cornerstone of ensuring the system's security and reliability. Through this process, we ensure that all communications and data exchanges during the Federated Learning process occur in a secure environment, thereby providing robust protection for user privacy.

ISSN: 2664-9640

The central server acts as the hub for model aggregation, with its core mission being the collection of encrypted gradient updates from various user terminals, execution of precise security aggregation operations, and generation of proofs for verifying the correctness of the aggregated results. This step is a key link in ensuring data consistency and accuracy during the model training process and is also an important safeguard for enhancing the overall system performance.

User terminals are responsible for training models using local datasets. In this process, gradients generated by users are encrypted and securely uploaded to the central server. This encryption mechanism not only protects user privacy but also ensures the security of data during transmission. Additionally, users are required to verify the aggregated results returned by the central server, a step that further enhances the system's security and user trust.

To further enhance system security, we have introduced random seed generation and One-Time Password (OTP) technology into our scheme. The application of these technologies significantly strengthens the security of data during transmission and processing. Concurrently, we have adopted an improved masking mechanism and bilinear pairing-based verification methods, which provide additional security for the model aggregation process. Most importantly, our scheme takes into account the flexibility needs of users, supporting their ability to exit the training process at will without adversely affecting other users or the overall training outcome of the model. This design not only reflects respect for user autonomy but also demonstrates the high adaptability and robustness of our scheme.

## 3. Specific Scheme

In this section, we will introduce the various stages and specific execution procedures of our proposed scheme.

System Initialization Phase. During this phase, the Trusted Third Party (TA) is responsible for generating the global parameters required by the system, including a random seed for subsequent encryption and verification processes. This random seed enhances the overall system security as it is utilized in subsequent encryption and verification operations, ensuring the randomness and unpredictability of each operation. The TA also generates a pair of public and private keys for each user. These keys are distributed to the users through a secure communication channel, and the public keys of all users are broadcast to facilitate their use in subsequent aggregation processes. Additionally, the central server initializes the parameters of a machine learning model and sends them to all users participating in the training.

Key Sharing Phase. In this phase, each user employs an improved Shamir's Secret Sharing technique, combined with a One-Time Pad (OTP), to segment and encrypt the random numbers used for gradient masking and the private keys for signing. This combined approach enhances the security of key sharing, as OTP provides perfect forward secrecy. Users transmit the encrypted ciphertexts to other users through the central server, which acts solely as a transmission intermediary without accessing the content of the ciphertexts.

Local Training and Masking Input Phase. Users train the model locally using their own datasets. After obtaining the gradients, they employ an improved masking mechanism to encrypt the gradients. This improvement may include the introduction of additional random factors to increase the complexity and security of the gradient masking. Users also calculate a signature that will be used to verify the correctness of the aggregated results. After completing these steps, users send the gradient ciphertexts and signatures to the central server.

Decoding Aggregation Phase. In this phase, users first receive key fragment ciphertexts from other users and then compute the key values used to decode the aggregated gradients. To ensure the correctness of the key values, a verification process is introduced, which may involve a challenge-response mechanism to verify the computed key values. All online users also need

ISSN: 2664-9640

to send the shared values of the signing private keys of users who went offline in the previous phase to the central server. The central server aggregates the gradient ciphertexts and verification signatures, calculates the decoded gradients (aggregated results) based on the decoded key values, and introduces an additional verification mechanism to verify the correctness of the aggregated results. Finally, the central server sends the decoded aggregated gradients and verification proofs back to all users.

Verification Phase. During this phase, after receiving the aggregated results and verification proofs sent by the central server, users conduct stricter verification calculations. This includes dual verification of the consistency and correctness of the aggregated results. Consistency verification ensures that all users see the same aggregated results, while correctness verification ensures that the aggregated results are indeed calculated based on the correct gradients. If the verification passes, users will continue to participate in the next round of iterative training; if the verification fails, users can choose to resend data or exit the entire training protocol.

## 4. Experimental Analysis

In this section, we comprehensively verify the effectiveness of our proposed scheme through the analysis of computational and communication overheads. Since the encryption and decryption processes in this paper do not affect the final performance of model training, the assessment of model performance is omitted, focusing instead on the performance evaluation of cryptographic operations.

### 4.1. Computational Overhead

This subsection focuses on analyzing the computational overhead of cryptographic operations in our scheme, excluding the time for user model training. We compare our scheme with two existing schemes (Versa[3] and VeriFL[4]) to demonstrate its performance.

As shown in Fig. 2, we compared the computational overhead for single-user verification across the three schemes as the number of users varies. The results indicate that our scheme has a significantly lower computational overhead for single-user verification, which is negligible and does not change with the number of users. Versa's overhead is slightly higher than our scheme, while VeriFL's overhead for single-user verification increases with the number of users. Verification in our scheme requires only three bilinear pairing operations, whereas Versa requires Hadamard product and multiplication operations on the gradient vector, and VeriFL requires multiplication operations on the gradient hash values of all users. The results demonstrate that our scheme is efficient for individual user verification.

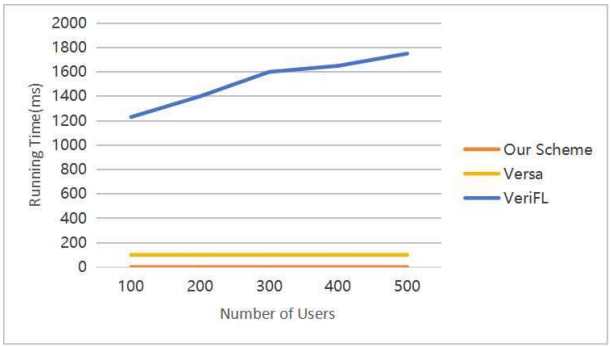


**Fig. 2** Computational Overhead for Single-User Verification Versus Number of Users Across Different Schemes

ISSN: 2664-9640

As depicted in Fig. 3, we observed that the total computational overhead for individual users slightly decreases with an increased dropout rate but increases with the total number of users. This is because, during the decoding aggregation phase, single-mask encryption only needs to calculate the secret sharing shares for online users, thus the more users drop out, the fewer shares need to be calculated. However, as the number of users increases, individual users need to encrypt and decrypt the ciphertexts of all users during the key sharing phase and decoding aggregation phase, leading to a significant increase in total computational overhead.

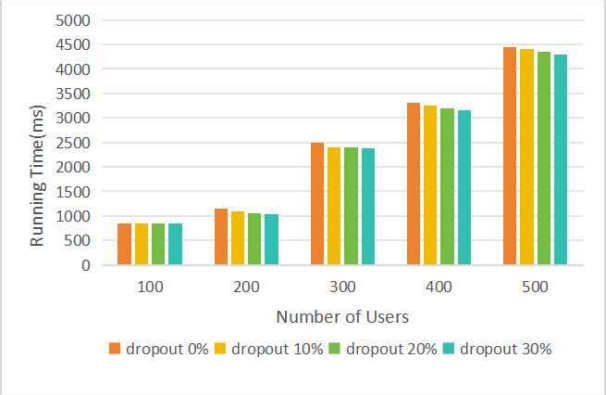


**Fig. 3** Computational Overhead for Single Users Versus Number of Users at Various Dropout Rates

Fig. 4 clearly shows how the server's computational overhead varies with the number of users under different dropout rates. It is evident that the server's computational overhead increases with the dropout rate, especially when there are more users and a higher dropout rate, the computational overhead increases significantly. This is because the server needs to reconstruct the verification private keys of users who have dropped out to pass the verification process; the more users drop out, the greater the computational overhead. Since servers typically have stronger computational resources compared to the limited resources of users, our scheme shows strong adaptability in handling a large number of user dropouts.

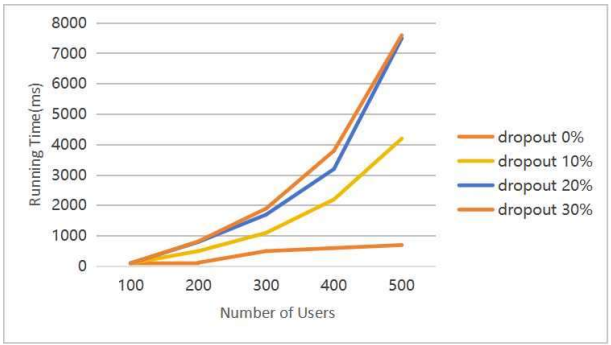


**Fig. 4** Computational Overhead for the Server Versus Number of Users at Various Dropout Rates

## 4.2. Communication Overhead

This subsection delves into the specific impact of user dropout rates on the communication overhead of our scheme.

Observing Fig. 5, the total communication overhead for individual users slightly increases with an increased dropout rate, indicating that an increase in the user dropout rate has a minimal impact on the total communication overhead for individual users. As shown in Figs 5(a) and

ISSN: 2664-9640

5(b), the total communication overhead for individual users increases less with the number of users compared to the increase with the number of gradients. This is because transmitting the masked gradient vector requires a larger overhead, which is an indispensable part of the privacy-preserving secure aggregation Federated Learning scheme. This indicates that our scheme will not significantly increase communication load with an increase in the number of users when the number of gradients is fixed.

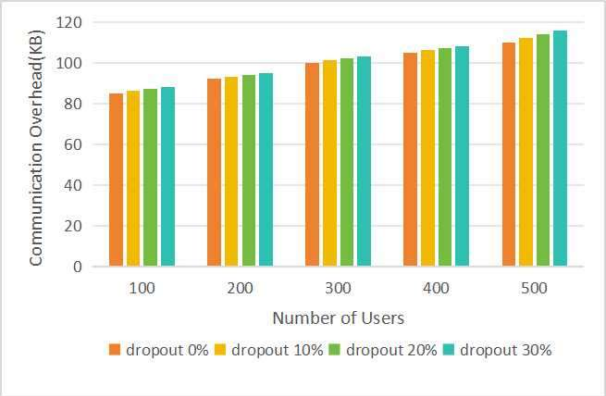


**Fig. 5(a)** Communication Overhead for Individual Users Varying with User Count at Different Dropout Rates

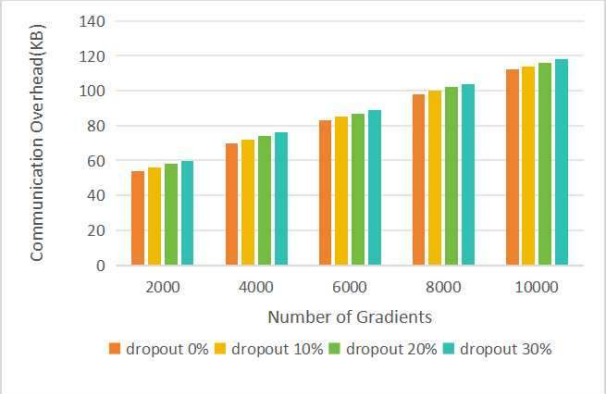


**Fig. 5(b)** Communication Overhead for Individual Users Varying with Gradient Count at Different Dropout Rates

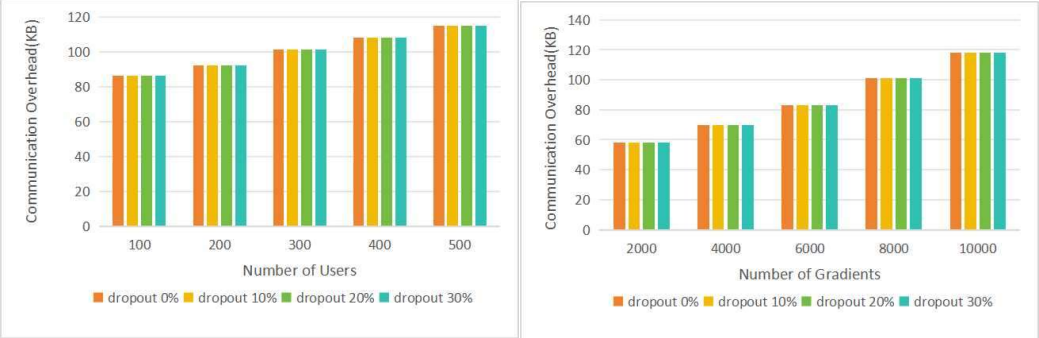


**Fig. 6** Communication Overhead for the Server Varying with User Count and Gradient Count at Different Dropout Rates

Fig. 6 illustrates how the server's communication overhead changes with the number of users and gradients under different dropout rates. It is evident that the server's communication

ISSN: 2664-9640

overhead is independent of the user dropout rate but increases with the increase in the number of users and gradients. The results from Figs 5 and 6 indicate that the user dropout rate has little impact on the communication overhead of our scheme, thereby confirming the high efficiency of our scheme in terms of user dropout tolerance.

### 4.3. Experimental Conclusion

The experimental results demonstrate that our scheme, while enhancing security, moderately increases computational and communication overheads. These increases are within an acceptable range and do not significantly affect the overall performance of Federated Learning. Especially when the user dropout rate is high, the scheme can still maintain low overhead, showing good robustness and scalability.

## 5. Summary

This paper introduces an enhanced Federated Learning scheme that prioritizes user privacy through innovative cryptographic techniques, including the use of random seeds and One-Time Passwords (OTP). The scheme ensures secure data transmission and robust model training, even in the face of user dropouts. Our experimental results confirm the scheme's efficiency, with minimal impact on computational and communication overheads, highlighting its practicality and robustness. The research significantly advances privacy preservation in FL, providing a solid foundation for future technological developments. The proposed scheme's ability to maintain functionality under user dropouts addresses a key challenge in distributed learning environments, showcasing its potential for real-world applications.

## References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54 of Proceedings of Machine Learning Research. PMLR, 2017.

[2] H. Brendan McMahan, Sarvar Patel, Daniel Ramage, et al. Karn Seth. Practical secure aggregation for privacy-preserving machine learning. CCS'17, pages 1175-1191, New York, NY, USA, 2017.

[3] Hahn C, Kim H, Kim M, et al. Versa: Verifiable secure aggregation for cross-device federated learning[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 20(1): 36-52.

[4] Guo X, Liu Z, Li J, et al. Verifl: Communication-efficient and fast verifiable aggregation for federated learning[J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 1736-1751.