A Smartnic-Based Secure Aggregation Scheme for Federated Learning

Shengyin Zang¹, Jiawei Fei², Xiaodong Ren¹, Yan Wang¹, Zhuang Cao^{3,*}, Jiagui Wu^{4,*}

¹College of Artificial Intelligence, Southwest University, Chongqing, China
 ²Defense innovation institute, Beijing, China
 ³College of Computer, National University of Defense Technology, Changsha, China
 ⁴School of Physical Science and Technology, Southwest University, Chongqing, China

Abstract

Federated learning is a widely used distributed machine learning technique where participants collaborate to train neural network models without disclosing private training datasets. Technologies like homomorphic encryption are proposed to avoid private data leakage. However, they are plagued by two problems, including performance degradation caused by excessive computational and communication overhead and insecurity due to the data exposure on the parameter server. Focusing on these problems, we propose an efficient and privacy-preserving federated learning solution that improves performance and security by offloading the aggregation procedure into the hardware data plane, like FPGA-based SmartNICs. Furthermore, we also combine it with differential privacy techniques. Our method has higher security because it is hard to access the data plane. Besides, massive experiments show that, compared to the system employing additively homomorphic encryption, our scheme reduces the communication cost by around 59.5% and offers around 2.5× speedup at the aggregation stage while significantly decreasing the participant's computational overhead.

Keywords

Federated Learning, Differential Privacy, SmartNIC, Secure Aggregation, Privacy-Preserving

1. Introduction

In the current era of big data, dispersed data cannot be integrated and utilized due to data security, privacy protection, and others, making data silos a severe obstacle to the advancement of artificial intelligence [1] [2]. In order to overcome this dilemma, federated learning technology [3] [4] was proposed by Google Research and is now widely employed in various scenarios. Federated learning is a distributed machine learning technique that enables model training on a sizable amount of decentralized data. The goal of federated learning is to accomplish multi-party cooperative modeling while safeguarding data privacy and satisfying regulatory compliance requirements. Compared with conventional machine learning techniques, federated learning not only improves learning efficiency but also resolves the problem of data silos and preserves local data privacy. However, it still runs the danger of leaking privacy information because it does not offer comprehensive and adequate privacy protection by itself. For instance, a malicious server can still deduce sensitive user information from the local gradients during the aggregation phase [5] [6]. To cope with this challenge, many scholars have conducted research on privacy leakage in federated learning. Homomorphic encryption [7] is one of the most commonly applied privacy technologies in federated learning. Although it offers security advantages, the tremendous computational complexity and ciphertext expansion also place a heavy burden on participants in federated learning in terms of computing and communication. Despite numerous optimization strategies for this issue [8] [9] [10], the performance bottlenecks remain fundamentally unresolved. Additionally, since the parameter server has access to ciphertext data, user privacy still faces the risk of leakage. Therefore, how to design a scheme to meet the above challenges re-

EMAIL: *caozhuang16@nudt.edu.cn (Zhuang Cao), mgh@swu.edu.cn (Jiagui Wu).



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CEUR Workshop Proceedings (CEUR-WS.org)

ICCEIC2022@3rd International Conference on Computer Engineering and Intelligent Control

mains an urgent problem. In this paper, we present an efficient federated learning privacy computing scheme based on hardware security by offloading the gradients aggregation operation to an FPGA-based SmartNIC [11] and combining it with differential privacy techniques [12] as an alternative to traditional software protections.

Our contributions are summarized as follows:

We propose a SmartNIC-based gradients aggregation algorithm, which improves the security and performance of federated learning by offloading the gradients aggregation operation onto SmartNIC.

We implement an efficient aggregation structure on SmartNIC, which enables our scheme to complete privacy-preserving computations while ensuring computational efficiency.

We evaluate the performance benefits of the SmartNIC-based gradients aggregation algorithm. Extensive experiments show that our scheme has lower communication and computational overhead than schemes using additively homomorphic encryption.

2. System model and threat model

2.1. System model

As shown in Figure 1, the SmartNIC-based federated learning system mainly consists of two main components: users and the SmartNIC-based parameter server on the cloud server. All users agree on an identical initial model and common training objectives. During the training process, all participants do not directly share their respective private data.



Figure 1 System model

On the users' side, they first download the global network model and the initial parameters from the cloud server, then perform model training based on the local dataset to obtain local gradients, encrypt and upload those gradients to the server, and finally decrypt the global gradients returned by the server to update local model parameters.

On the SmartNIC-based parameter server, the primary task of SmartNIC is to achieve efficient and secure aggregation of gradients by utilizing its hardware-isolated execution environment and extreme computing performance. The SmartNIC decrypts local gradients uploaded by each participant, aggregates them to obtain global gradients, and adds Gaussian noise perturbation to thwart differential attacks [13]. The global gradients are then broadcast to all users after being encrypted. After continuous iterations, until the loss function reaches the minimum value, the optimal neural network model is finally constructed.

2.2. Threat model

Our proposed scheme aims to safeguard the user's private information throughout the training phase. The cloud server is assumed to be honest-but-curious, meaning that it will adhere to the proto-

col to execute gradients aggregation, while also showing some degree of curiosity about the users' raw data and may attempt to bypass some security measures to access the users' privacy data directly. Additionally, malicious participants try to determine whether a particular user is involved in the training process by analyzing the shared global gradients, i.e., performing membership inference attacks [14].

3. Proposed scheme

In this section, we propose a SmartNIC-based federated learning gradients aggregation scheme that works as an alternative to traditional software protections. The fundamental idea is to provide a trusted execution environment [15] for isolated execution of procedures and privacy data processing by offloading the gradients aggregation operation to the FPGA-based SmartNIC. It will be difficult for the server to access the data in the SmartNIC to realize the privacy calculation of sensitive data. In addition, the perturbation noise satisfying the Gaussian mechanism is added to the aggregated global gradients to prevent user privacy data from being subjected to differential attacks and model overfitting, although this may lead to a decrease in model accuracy or increase the number of iterations required for model convergence.

3.1. Overview of secure aggregation scheme

Figure 2 illustrates the proposed aggregation scheme, and the process of joint modeling can be broadly divided into the following parts:



Figure 2 Schematic of aggregation scheme

Initialization. The cloud server broadcasts the global network model, the initial parameters ω_0 of the model and the learning rate η to all users participating in the training. Different encryption methods will be assigned to each participant at the same time.

Local Training. Based on the network model and initial parameters distributed by the server, each participant trains based on stochastic gradient descent [16] with a mini-batch of local datasets and calculates the local gradients. Those gradients are then encrypted and sent to the server's SmartNIC for aggregation.

Secure Aggregation. SmartNIC separately decrypts local gradients from different participants. After receiving all users' data, SmartNIC starts to perform aggregation operations. The aggregated global gradients are perturbed for privacy-preserving purposes with noise satisfying a Gaussian distribution. Finally, those gradients are broadcast to all users after being encrypted by the SmartNIC.

Global Update. After receiving the global gradients returned by SmartNIC, the user decrypts them and updates the model parameter ω according to the global gradients and learning rate η .

The system will repeat the above steps until the loss function reaches the minimum value. The final neural network model is constructed through a continuous loop iteration between SmartNIC and users. During the whole training process, the server only manages the SmartNIC but cannot access the users' private data on the SmartNIC, thus playing the role of privacy protection.

3.2. Aggregation architecture on SmartNIC

Figure 3 outlines the aggregation structure on the SmartNIC. During the entire training process, the data handling is wholly implemented on the SmartNIC, while the cloud server only configures the SmartNIC in the initial stage of training. For input transactions, the Ethernet MAC parses the data from the Ethernet PHY and transmits it to the Decrypt Engine for decryption. The decrypted user data will be stored in the DDR of the Storage Engine, where the DDR allocates separate memory space for each user. Once all the memory spaces contain user data, the Aggregate Engine initiates the gradients aggregation operation and transfers the resulting data to the Perturb Engine for perturbation, where the noise satisfies the Gaussian distribution. The disturbed global gradients are then encrypted by the Encrypt Engine and stored in the FIFO. For output transactions, once a complete packet is stored in the FIFO, the Ethernet MAC drives the Ethernet PHY to distribute the encrypted global gradients to all users. Both transactions use the standard 128-bit AXI-stream bus to interact with the Ethernet interface.



Figure 3 Architecture of aggregation scheme

4. Implementation

In this section, we will then discuss the hardware structure of the components on the SmartNIC.

Decrypt Engine. We have implemented DES and 3DES encryption and decryption algorithms in a pipelined manner on the decryption engine. After receiving the data, the decryption engine will decrypt it based on the user ID and sequence number. The Encrypt Engine uses a similar structure for encryption.

Storage Engine. The structure of the Storage Engine is shown in Figure 4. Each user is assigned a separate memory space in the DDR of the Storage Engine. When data enters the Storage Engine, the user ID and sequence number are used to control where the data is stored and to check for packet loss. Data is read from the DDR and delivered to the aggregation engine after being synchronized by the FIFOs when none of the address spaces are empty.





Aggregate Engine. The structure of the Aggregate Engine is shown in Figure 5. We implement a scenario with up to 64 sets of gradient data that can be aggregated simultaneously, which uses a six-stage pipeline structure to improve the processing performance. 32 Aggregate Blocks (ABs) are used in the first stage, and each AB aggregates two gradients. 16 ABs are used in the second stage, and then the number of ABs is halved sequentially in subsequent pipelines until the final global gradients are output and submitted to the Perturb Engine. In the case of more than 64 users, the result should be temporarily stored in the DDR pending further aggregation. Moreover, if fewer than 64 users, set some of the ABs' inputs in the pipeline's first stage to zero.



Figure 5 64-bit Aggregate Engine architecture

The structure of the Aggregate Block is shown in Figure 6. Since the Encrypt Engine only implemented DES and 3DES encryption algorithms and the ciphertexts are 64 bits, we presume that the input of the AB is also 64 bits. Gradients data are presented as 32-bit fixed-point numbers. The input of AB is the gradient data after 32-bit alignment, i.e., two gradients at a time. AB splits the input and sends it to the two adders, respectively. The results of the two adders are concatenated into 64 bits and then delivered to the register for temporary storage.



Figure 6 64-bit Aggregate Block architecture

5. Experiment

In this section, we will assess our proposed scheme from the aspects of communication and computational overhead, and hardware resource consumption by comparing it with the novel scheme PPDL [8], in which additively homomorphic encryption is adopted as the privacy-preserving approach. We implemented the SmartNIC-based secure aggregation structure on the Xilinx Zynq UltraScale+ZCU111 evaluation platform using Verilog language, and used Vivado 2018.3 for logic synthesis and implementation. Without path violations, the final data processing speed can be up to 425 MHz, and the data throughput rate can reach approximately 27 Gbps. Table 1 describes the resource utilization of the SmartNIC-based aggregation scheme. PPDL's benchmarks are based on the Tensorflow 1.1.0 library over Cuda-8.0 and GPU Tesla K40m, with a Xeon CPU E5-2660 v3@ 2.60GHz server, and assume that each user uses only one thread. Since we completely offload the gradients aggregation process in federated learning from the cloud server to the SmartNIC, and the cloud server only configures the SmartNIC, the resource utilization on the cloud server of our scheme can be approximately considered zero compared with that of PPDL. Next, we will mainly compare the two schemes' communication and computational overhead differences.

Resource	Utilization	Available	Utilization %
LUT	40962	425280	9.63
LUTRAM	663	213600	0.3111
FF	30366	850560	3.57
BRAM	256	1080	23.70
DSP	2	4272	0.05
BUFG	1	696	0.14

Table 1 The area utilization report of our scheme

5.1. Communication overhead

Assuming that each participant uses only one thread for computation, we first compare the communication overhead of each participant in federated learning in PPDL. The relationship between the communication overhead and the number of gradients is shown in Figure 7. The figure clearly indicates that the communication overhead of PPDL is more than twice as high as our scheme. The main reason is the rapid growth of ciphertext volume brought on by homomorphic encryption.



Figure 7 Comparison of communication overhead

5.2. Computational overhead

The computational cost differences between our approach and PPDL during the encryption, aggregation, and decryption stages will be compared. Since each participant in our scheme may use different encryption methods, we pick the 3DES algorithm with the highest computational cost to compare with the homomorphic encryption of PPDL. As shown in Figure 8, our encryption overhead is considerably lower than that of PPDL as the number of gradients increases, which is caused by the high computational complexity of homomorphic encryption. Figure 9 describes the difference in computational cost between our scheme and PPDL in the aggregation stage. It is worth noting that the aggregation stage in our scheme includes four sub-stages: decryption, aggregation, noise addition, and encryption. Benefiting from the high performance of SmartNIC, our computational overhead in the aggregation phase is less than half of PPDL. Similarly, as shown in Figure 10, as the number of gradients increases, the decryption overhead is also much smaller than PPDL. Therefore, our scheme is more suitable for training large-scale deep neural network models.



Figure 8 Comparison of computational overhead at the encryption phase



Figure 9 Comparison of computational overhead at the aggregation phase



Figure 10 Comparison of computational overhead at the decryption phase

6. Related work

Some related work on increasing the performance and security of federated learning systems by using homomorphic encryption has been studied recently. For example, AONO Y et al. [8] implemented the aggregation of gradients on the cloud server using an additively homomorphic encryption scheme with low computational load and guaranteed the system's high accuracy. Even yet, there is still a substantial computational overhead associated with this scheme due to the growing size of the neural network and the growing amount of training set samples involved in the modeling process. At the same time, this scheme is also vulnerable to differential attacks, and the privacy of honest participants' data will be threatened by the analysis of the shared model [17]. For this reason, Meng Hao et al. [9] proposed an efficient federated deep learning scheme by integrating a lightweight symmetric additively homomorphic encryption [10] with differential privacy [12]. This scheme is secure for honest-but-curious server settings, even if the cloud server colludes with multiple users. Unfortunately, all participants in this scheme use a single encryption key, making the system vulnerable to security threats even though the aggregation can be carried out smoothly. Therefore, how to design a scheme to meet the above challenges remains an urgent problem.

7. Conclusion

This paper proposes an efficient and privacy-preserving federated learning system based on hardware security by offloading the gradients aggregation operation onto SmartNIC as an alternative to homomorphic cryptography. Extensive experiments demonstrate that our scheme has lower communication and computational overhead than schemes using additively homomorphic encryption. Meanwhile, our scheme is secure for the honest-but-curious cloud server and has superior security by using different encryption methods for each participant compared to single-key homomorphic encryption. In addition, we further add Gaussian noise that satisfies differential privacy to the aggregated global gradients to defend against differential attacks in federated learning. In future work, we will train realworld deep neural networks to evaluate the impact of our scheme on model accuracy compared to traditional centralized machine learning and investigate using SmartNIC clusters for secure aggregation to support federated learning for larger-scale applications.

8. Acknowledgment

This work is supported by the National Natural Science Foundation of China (61875168); Chongqing Science Funds for Distinguished Young Scientists(cstc2021jcyj-jqX0027); Innovation Research 2035 Pilot Plan of Southwest University (SWU-XDPY22012); Innovation Support Program for Overseas Students in Chongqing (cx2021008).

9. References

- T. Tiffany, L. Josh, M. Daniele, "Asynchronous collaborative learning across data silos," arXiv preprint arXiv:2203.12637, 2021.
- [2] P. Franz, S. Olga, R. Kay, G. Florian, J. Igor, S. Franz et al., "Embracing opportunities of livestock big data integration with privacy constraints," in Proceedings of the 9th international conference on the internet of things, pp. 1-4, 2019.
- [3] H. B. McMahan, E. Moore, D. Ramage, B. A. Arcas, "Federated learning of deep networks using model averaging," arXiv preprint arXiv:1602.05629, 2016.
- [4] Y. Cheng, Y. Liu, T. Chen, Q. Yang, "Federated learning for privacy-preserving AI," Communications of the ACM, vol. 63, no. 12, pp. 33-36, 2020.
- [5] Z. Ligeng, L. Zhijian, H. Song, "Deep Leakage from Gradients," arXiv preprint arXiv:1906.08935, 2019.
- [6] X. Xiaoyun, W. Jingzheng, Y. Mutian, L. Tianyue, D. Xu, L. Weiheng Li et al., "Information Leakage by Model Weights on Federated Learning," in Proceedings of the 2020 workshop on privacy-preserving machine learning in practice, pp. 31–36, 2020.
- [7] M. Naehrig, K. Lauter, V VaikuntanathanN, "Can homomorphic encryption be practical?," in Proceedings of the 3rd ACM workshop on cloud computing security workshop, pp. 113–124, 2011.
- [8] Y. Aono, T. Hayashi, L. Wang, S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," IEEE transactions on information forensics and security, vol. 13, no. 5, pp. 1333–1345, 2018.
- [9] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in 2019 IEEE international conference on communications, pp. 1–6, 2019.
- [10] Z. Jun, C. Zhenfu, D. Xiaolei, and L. Xiaodong, "Ppdm: A privacy-preserving protocol for cloud-assisted e-healthcare systems," IEEE Journal of Selected Topics in Signal Processing, vol. 9, no. 7, pp. 1332–1344, 2015.
- [11] L. Ming, C.Tianyi, S. Henry, K. Arvind, P. Simon, G. Karan, "Offloading distributed applications onto smartNICs using iPipe," in Proceedings of the ACM special interest group on data communication. Association for Computing Machinery, pp. 318–333, 2019.
- [12] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar et al., "Deep learning with differential privacy," in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, 2016, pp. 308–318.
- [13] Z. Chuanxin, S. Yi,W. Degang, "Federated Learning with Gaussian Differential Privacy," in Proceedings of the 2020 2nd international conference on robotics, intelligent control and artificial intelligence, pp. 296–301, 2020.
- [14] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017 IEEE symposium on security and privacy (SP), 2017, pp. 3-18.
- [15] A. Mondal, Y. More, R. H. Rooparaghunath, D. Gupta, "Flatee: Federated Learning Across Trusted Execution Environments," arXiv preprint arXiv:2111.06867, 2021.

- [16] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, 2015, pp. 1310– 1321.
- [17] R. Geyer, T. Klein, M. Nabi, "Differentially private federated learning: a client level perspective," arXiv preprint arXiv:1712.07557, 2017.