Distributed Learning in Trusted Execution Environment: A Case Study of Federated Learning in SGX

Tianxing Xu[1], Konglin Zhu [1], Artur Andrzejak [2], Lin Zhang [1]

[1]School of Artificial Intelligence, BUPT, Beijing 100876, China.
[2] Heidelberg University, Heidelberg 69120, Germany.
xutx@bupt.edu.cn

Abstract: Federated Learning (FL) is a distributed machine learning paradigm to solve isolated data island problems under privacy constraints. Recent works reveal that FL still exists security problems in which attackers can infer private data from gradients. In this paper, we propose a distributed FL framework in Trusted Execution Environment (TEE) to protect gradients in the perspective of hardware. We use trusted Software Guard eXtensions (SGX) as an instance to implement the FL, and proposed an SGX-FL framework. Firstly, to break through the limitation of physical memory space in SGX and meanwhile preserve the privacy, we leverage a gradient filtering mechanism to obtain the "important" gradients which preserve the utmost data privacy and put them into SGX. Secondly, to enhance the global adhesion of gradients so that the important gradients can be aggregated at maximum, a grouping method is carried out to put the most appropriate number of members into one group. Finally, to keep the accuracy of the FL model, the secondary gradients of group members and aggregated important gradients are simultaneously uploaded to the server and the computation procedure is validated by the integrity method of SGX. The evaluation results show that the proposed SGX-FL reduces the computation cost by 19 times compared with the existing approaches.

Keywords: Federated Learning; Privacy preserving; Trusted execution environment

1 Introduction

Federated learning [3] has recently emerged as an important setting for distributed machine learning models training. In this paradigm, edge users participate in the model training while retaining their local training data and upload only the local model updates (i.e., weight and gradients parameters) to the cloud server. Those local models' updates are then aggregated and processed to obtain a new global model on the server. This cycle is repeated until the desired learning accuracy is achieved. By sharing gradients instead of sensitive data, the FL is considered to be secured and privacy preserved.

However, some recent studies show that gradients reveal the training data privacy. [7] proposed a GAN-based reconstruction attack against collaborative learning by assuming a malicious server, which utilized the shared model as the discriminator to train a GAN to mimic the training samples. Zhu et al. [1] proved the possibility of obtaining private training data from publicly shared gradients, known as Deep Leak from Gradients. Such

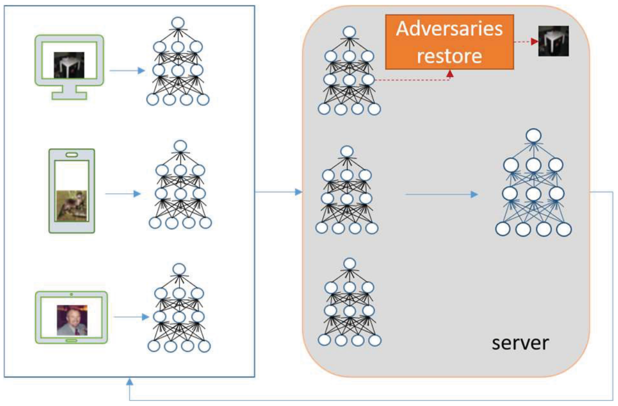

Figure 1 The data leakage from gradients in FL.

kind of data leakage from gradients can also be exploited by adversaries in FL to recover the data from local device resulting in the corrupted data privacy, as shown in Fig. 1.

Indeed, securing the data for FL is not an easy task and faces the following challenges. First of all, the extent of data security been protected is limited. Many approaches [2][4] discussed the FL security in aspect of data and model security while neglecting the integrity of the data, thus involving malicious attacks from inside users. Secondly, data protection is still not efficient enough to meet the requirements of FL. The approaches such as Secure multi-party computing (SMC) [6] and homomorphic encryption [7] are introduced for privacy preserving in FL. However, their computation efficiency is very restricted. Thirdly, the accuracy of the model is sacrificed when the FL adapts to the security frameworks. Differential Privacy [8] based techniques add noise to data in order to protect individual privacy. However, this noise reduces data utility, and makes model performance deterioration. Some works studied to compress the model [10] for secure and efficient computation [2], which usually needs to sacrifice the quality.

In this paper we address the challenges mentioned above and present a secure FL framework, named SGX-FL. The proposed framework reduces the computation cost, maximizes the performance of model and protects privacy from being compromised by adversaries. In particular, we first apply a filtering mechanism for models on each client to obtain the "important" gradients which the weights are utmost to adapt to the limited memory of SGX. Then we group and upload "important" gradients in order to increase the adhesion so that the important gradients can be aggregated in SGX of group leader. Finally, to keep the accuracy of the model, the secondary gradients of group members and aggregated important gradients are simultaneously uploaded to the server cloud by group leader and the computation procedure is validated by the integrity method based on SGX.

The major contributions in this paper are summarized as follows:

- We propose an SGX-FL, a secure distributed learning framework that simultaneously reduces computation cost, maximizes the performance of model and protects privacy from being compromised by adversaries.

- We carry out the gradients filtering and "important" gradients grouping method as well as data integrity verification approach to adapt to the limited memory of SGX and improve computation cost.

- Extensive evaluation shows that the proposed SGX-FL framework can effectively protect data models so that the data cannot be recovered and leaked. The results present that the proposed framework outperforms the state-of-the-art in terms of computation cost by 19 times.

2 Related Works

2.1 Non-TEE based FL model protection

The existing Non-TEE based FL model protection needs to be improved in terms of model accuracy and/or efficiency. Secure multi-party computing (SMC) [4] protocol based on secret sharing, called security aggregation. Their lower computational overhead is abstract. However, garbled Circuit based solutions need complex circuit design and optimization, which limit its flexibility and usability greatly. Homomorphic encryption [6] is used to aggregate gradients, but cryptographic operations require a large amount of computing resources. Differential Privacy [7] based techniques add noise to data in order to protect individual privacy. But, this noise reduces data utility, and makes model performance deterioration.

2.2 TEE-based FL model protection

To avoid the above threat model, in this paper, we use TEE provided by Intel Software Guard eXtensions (SGX). Intel SGX is a set of extensions to the Intel architecture for secure remote execution. In particular, Intel SGX protects the confidentiality and integrity of code and data through creating a protected physical memory known as an enclave. However, the size of the memory area of the enclave is limited to 128 MB. This makes the performance of FL training in SGX is poor. The basic CNN model AlexNet requires 260 MB of memory, while the fashion used model VGG-19 requires approximately 1 GB of memory space to load model weights and compute convolution.

[4] adopt the strategy of hierarchical aggregation which effectively solved the problem of insufficient memory of SGX. But due to serial strategy, the computation cost of server aggregation was unacceptable. [1] proposes that deep gradients leakage problem can be solved by sparseness, but the threshold of 20% still bring high computational cost. [2] is performed by protecting a portion of the gradients, however, at the expense of model performance. Impose severe performance challenges led prior works to forgo supporting DNN models or a full protection of data privacy [11].

3 SGX-FL Framework

3.1 Threat model

We consider a scenario where some local clients desire to obtain a high-quality deep neural network (DNN) model yet without substantial training data. We suppose that the local clients are honest-but-curious attackers in the FL. The meaning of honest-but-curious is that the local clients completed the FL task correctly to obtain a high-quality DNN model, but attempt to infer the data privacy of other local clients during the global model update. Federated Learning framework eliminates the threat of breaching data privacy directly. However, DLG [1] can indirectly steal raw training data from gradients.

The server cloud plays a role that coordinates local clients to aggregate training results. The server cloud is dishonest, which means the server cloud instances, including privileged software like OS and hypervisor, are not trusted. Attacks can be performed by the cloud provider or anyone with access to the OS/hypervisor. Such as attacks may 1) return arbitrary model updates without aggregation; 2) tamper with global model, i.e., change all the weights of certain layers or some weights of all layers [10].

Based on this supposition, we guarantee the privacy of local clients' data and the integrity of the aggregation results. In addition, we don't sacrifice too much computation cost. Although some studies also discussed the attacks against the TEE, these attacks are not considered in this paper.

3.2 System overview

Figure 2 shows the architecture of SGX-FL. The system is comprised of groups nodes and a server node. The execution of SGX-FL can be summarized as follows:

The local clients use their own data training to get the gradient. Then we apply a gradient filtering for model on each client to obtain the "important" gradients.

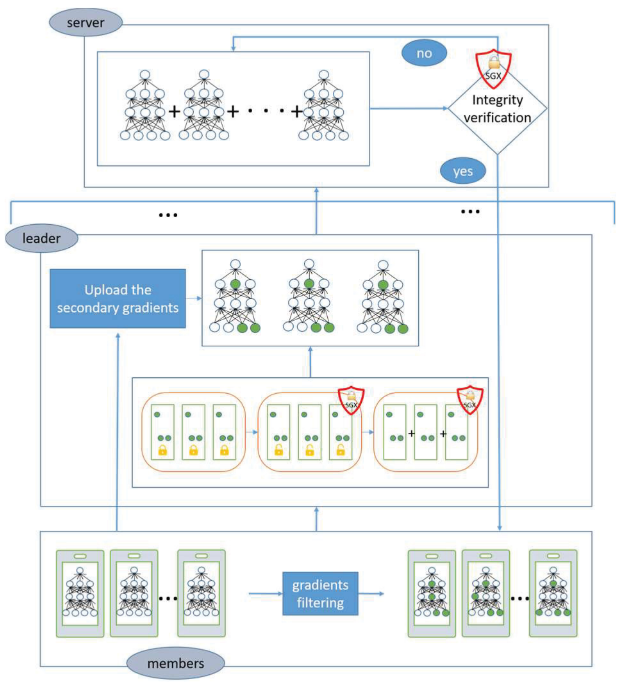

Figure 2 An overview of SGX-FL.

The grouping algorithm reorganizes local clients into a hierarchical group structure. The group members encrypt their important gradients, upload them to the SGX of group leader. The important gradients are decrypted in SGX after SGX authentication integrity is successful. We implement the FederatedAveraging (FedAvg) [3] according to the amount of data each group member had:

$$G^{t+1} = \eta \sum_{j=1}^{K} \frac{n_j}{n} G_j^t$$

where η is the global learning rate that instructs how much the model updated at every round, K is the collection of group members, n is the total data size, n_j and G_j^{t+1} respectively are the training data size and gradients of jth group member in K. Then the group members upload the secondary gradients to leader to ensure the accuracy of the model performance.

Each group leader uploads all the gradients of the group to server cloud. SGX performs validation to verify that the cloud is honestly performing the expected aggregation with a high probability guarantee. If the validation is successful, the cloud distributes the aggregated results to the local client. Otherwise, it restarts aggregation. If the global model converges, the cloud ends the learning task. Otherwise, it returns to the first stage.

With this system design, SGX-FL achieves all its design goals. With only protecting a part of gradients, SGX-FL avoids exceeding SGX memory and thus significantly reduces the computation cost. With grouping by adhesion, SGX-FL effective defense DLG and MIA. By data integrity verification, SGX-FL effectively protects the integrity of the aggregation in server.

4 Design details

We discuss the SGX-FL design details in this section. The SGX-FL framework mainly contains three major components: “important” gradient filtering, hierarchical grouping and data integrity verification.

4.1 “Important” gradient filtering

Because of the memory limitations of SGX, we cannot protect the entire machine learning model by SGX, especially deep learning model. If the whole DNN model is directly protected by SGX, it increases the computation cost due to the memory limitations of SGX. In order to reduce the computation cost under the premise of protecting data privacy, we need to protect a portion of gradients which is “important”. The importance is defined as the privacy revealed from the gradients. Prior work [1] reveals that privacy is revealed less when the gradient is filtered more, which means we need to select the high proportion gradients into SGX. Therefore, we calculate the gradients G which can be obtained through Stochastic Gradients Descent (SGD) optimization:

$$G_j^t = \frac{\partial}{\partial \mu^t} J$$

where μ is model parameters of jth local client in t round, G_j^t respectively is gradients of jth local client in t round, J is loss function.

Then we arrange the gradient from largest to smallest into a one-dimensional vector V_j. We defined the proportion of gradients selected as α of which value is determined by requirement of the privacy preserved and computation cost. We can calculate the threshold of filtering:

$$\text{threshold} = V_j[\alpha * \text{S}]$$

where S is the size of gradient. Finally, we can get the index of the important gradient in_j:

$$\text{in}_j[\text{a, b}] = \begin{cases} 1, & if\ gradient[a,b] > threthold \\ 0, & otherwise \end{cases}$$

We choose the size of the gradients as a vital indicator of our filtering mechanism because it's a variation of the degree which the model is updated in this round. The filtering mechanism can achieve better security and lower computational cost by protecting important gradients.

4.2 Hierarchical grouping

In a general FL structure, the server aggregates all the clients, which results in low adhesion. Adhesion represents the degree to which important gradients of different local clients overlap. For example, the adhesion between client i and client j is as follows:

$$\text{adhesion} = \frac{\sum\sum \text{in}_{i,j}[a,b]}{S}, \qquad \text{in}_{i,j} = \bigwedge \text{in}_i, \text{in}_j$$

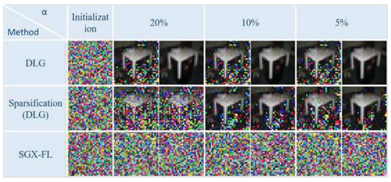

Figure 3 The visualization showing the deep leakage on images from CIFAR-100, when α ranges from 5% to 20%.

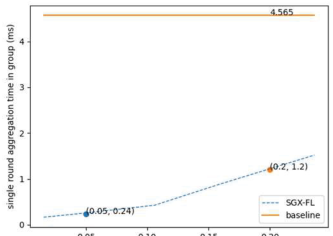

Figure 4 The cost of SGX training.

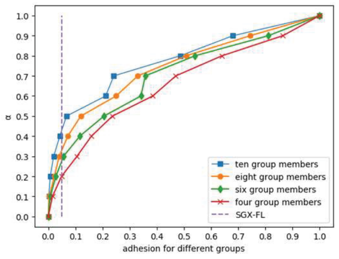

Figure 5 The adhesion, when the size of group members ranges from 4 to 10.

Adhesion is a gradient ratio influenced by two factors. Increasing α can improve adhesion. But it leads to the severe dilution of the important gradients and SGX-FL increasingly close to a random sparse strategy, which reduces security. In extreme cases, entire gradients are selected as the important gradients. At this point, the adhesion reaches the highest, and the safety of SGX-FL will be equal to random sparsity. Too many clients directly aggregate important gradients, causing the index of the important gradient to fail to align, which eventually results in low adhesion. We choose a hierarchical group structure to solve this problem. The server cloud manages all group leaders G. A group leader coordinates group member. Excessive adhesion leads to excessive gradient aggregation in SGX, which results in high calculation cost. Low adhesion leads to reduce security. We used a strategy to control the adhesion in $[\beta_1, \beta_2]$. Each local client generates a random value in $[1, G]$. Group members with the same values select the members with SGX and the largest computing resources as group leader. The server initializes an appropriate α. The group leader calculated the adhesion to control the size of the group. For example, when the adhesion is greater than β_2, we can increase the size of group until adhesion is in $[\beta_1, \beta_2]$.

4.3 Data integrity verification

A malicious server can compromise the integrity of data [10]. We designed the data integrity verification to ensure correct aggregate results. At the beginning of each round of aggregation, the server cloud sends a verification request to SGX. Note that the Verify integrity work is synchronized with the aggregation work, so there is no obvious additional time cost. SGX verifies the outside aggregation execution by checking random-selected neural network gradients. Specifically, the distrust server stores all model gradients at the end of each aggregation round, and sends a commit message to the TEE when aggregation finishes. SGX randomly aggregates a portion of the gradients m of each layer, which makes the detection probability for such misbehavior higher. The probability P equals to:

$$P = 1 - \frac{C_{M-m}^{a}}{C_{M}^{a}}$$

where M is the size of gradients in layer, a is the modified gradients of layer. For example, in the FC1 layer of LeNet, we can achieve 99% confidence by verifying only one percent of the gradients when the attacker wants to reduce the performance of the model by tampering with 1% of the gradients a. In the small layers (e.g., conv1, fc3), we can obtain a higher confidence P by adjusting the verification ratio appropriately. Our validation algorithm only needs to use high validation rates in small neural network layers. The calculation cost doesn't increase significantly in this case.

Table 1 Each layer of LeNet verifies integrity confidence.

	Conv	*Conv2*	*Fc1*	*Fc2*	*Fc3*
M(FLOPS)	4800	10240	39360	7392	193
m(FLOPS)	480	1024	3936	739	300
a(FLOPS)	48	1024	3936	739	20
P	99%	99%	99%	99%	97%

5 Experiments

5.1 Experimental setup

To make the results comparable, we use two publicly available datasets: MNIST and CIFAR-10, that train on the CNN model with LeNet in the simulations. In particular, the MNIST dataset contains 60,000 training samples where each sample is based on images of handwritten digits ranging from 0–9, i.e., (10 different classes) with a size of 28 × 28 pixels. Whereas the CIFAR-10 dataset contains 50,000 training samples and 10,000 testing samples where each sample consists of images of 32 × 32 pixels with three different RGB channels. We use an Intel Core i7-6700 Skylake 3.40GHz processor with 16GB of RAM, a desktop processor with SGX support. The distrust computations are performed on an NVIDIA GeForce RTX 1080 Ti GPU. Due to a lack of native internal multi-threading in SGX, we run our TEE on a single CPU thread.

5.2 Security of SGX-FL

We compare the DLG which uses a random sparsity strategy [9] and SGX-FL in terms of data privacy preserving. We visualize the data privacy preserving effects in Fig. 3. It shows that sparsification (DLG) recovers a small quantity local image when the α is 20%. But when the α is 20%, the DLG method almost fully recover the image. In SGX-FL, local image cannot be recovered by DLG whatever α is.

Table 2 The loss of the deep leakage on images from CIFAR-100, when α range from 5% to 20%.

Method	Loss			
	20%	10%	5%	0%
SGX-FL	50.3342	43.2846	26.6185	0
DLG [1]	0.1173	0.0047	0.0029	0

We also show the loss of the different methods in Table 2. It obviously shows that the loss of SGX-FL is significantly greater than DLG. This is why the image cannot be recovered in SGX-FL.

5.3 Computation cost of SGX-FL

We compare the running time of aggregating all models into SGX with ours. In the scheme that all models are aggregated in SGX, the SGX plays the role of aggregation center and aggregates fully gradients from group members in group leader's enclave. Fig. 4 shows each round model aggregation time when the number of group members is four. In the baseline scheme that all gradients are aggregated in SGX, the computation time is always 4.565 milliseconds. In SGX-FL, as our observation indicates, as little as 5% of local gradients are considered important, data privacy can be reserved. When 5% of local gradients are selected, the computation time for one round model aggregation is 0.24 milliseconds, which is 19 times faster than that of the baseline algorithm. Even 20% of local gradients are selected. The computation time for one round model aggregation is 1.2 milliseconds, which is still 3.8 times faster than that of the baseline algorithm. With the enhancing adhesion high computational cost becomes more and more until the adhesion to 100% computational cost is equal to the baseline. However, the security will be reduced due to the high group members' important gradients ratio.

5.4 Adhesion of SGX-FL

We evaluate that the group aggregation method is feasible by examining the adhesion. As shown in Fig. 5, with the growth of adhesion, α is increasing. For example, when the group size is 8, α to reach the requirement of SGX-FL is 30%. Higher important gradient rates result in reduced security. As the size of the group increased, the curve grows more and more sublinear. In SGX-FL, α equal to 20% was required for a 4-member group and 45% for a 10-member group. Excessive α will dilute the important gradients and reduce security, as it gets closer and closer to the strategy of random sparsity. This is consistent with 4.2 above. The hierarchical group structure can not only meet the needs of privacy protection, but also effectively reduce α.

5 Conclusion

In this paper, we use SGX as an instance to implement the FL, and proposed a secure SGX-FL framework. The proposed approach simultaneously reduces the computation cost, maximizes the performance of model and protects the privacy from being compromised by adversaries. We deploy a testbed supporting the proposed system for performance evaluation. Experimental results show that the proposed SGX-FL outperforms the state-of-the-art by 19 times in terms of computation speed.

Acknowledgements

This work is supported by Hunan province science and technology project funds(2018TP1036).

References

[1] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients.", Federated learning, Springer, Cham, 2020, pp.17-31.

[2] M. Asad, A. Moustafa, and M. Aslam, "CEEP-FL: A comprehensive approach for communication efficiency and enhanced privacy in federated learning.", Applied Soft Computing, 2021, Vol.104, p.107235.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 2017, pp.1273-1282.

[4] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "PPFL: privacy-preserving federated learning with trusted execution environments", 2021, arXiv preprint arXiv:2104.14380.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning.", In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, New York, NY, USA, 2017, pp.1175–1191.

[6] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption.", IEEE Transactions on Information Forensics and Security, 2018, Vol.13, pp.1333-1345.

[7] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning.", In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, New York, NY, USA, 2015, pp.1310–1321.

[8] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for CPU based attestation and sealing,", Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy, Association for Computing Machinery, New York, NY, USA, 2013, Vol. 13, p. 7.

[9] Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training.", 2017, arXiv preprint arXiv:1712.01887.

[10] T. Florian and D. Boneh. "Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware.", 2018, arXiv preprint arXiv abs/1806.03287.

[11] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving Machine Learning as a Service.", 2018, arXiv preprint arXiv:1803.05961.