Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/69/20241477

Fully homomorphic encryption in PPML: An review

Jingting Liu

Chongqing University, Chongqing, 400000, China

13370320356@163.com

Abstract. Fully homomorphic encryption (FHE) in privacy-preserving machine learning (PPML) is a current area of research value, aiming to achieve the protection of users' private data by applying the concept of full homomorphic encryption to machine learning privacy preservation. The integration of the two involves extensive model modifications and performance issues. The current difficulties mainly focus on how to improve encryption efficiency through hardware or software, and how to apply homomorphic encryption to neural network models such as RNN that process sequence data. This paper introduces this complex research field, outlines two machine learning service models (MLaas and AIaas) that are concerned by the industry, summarizes the most advanced research technologies based on these two models in recent years, and discusses the technical difficulties and future research directions. As a difficult problem that has never been overcome in cryptography in recent decades, homomorphic technology has received extensive attention from experts and scholars and ushered in new opportunities in the current explosive development of machine learning.

Keywords: homomorphic encryption, privacy-preserving, machine learning.

1. Introduction

With the wide application of machine learning in many fields, such as image recognition, self-driving cars, medical image segmentation, natural language processing, and content pushing, the amount of training data and intermediate results is getting huge as well as the problem of users' private data protection, which is becoming more and more prominent. Machine learning algorithms usually need to train and predict on a large amount of user data, and in this process, the users' private data (e.g. personal identity information, bank account information, facial picture information) may lead to privacy leakage and abuse. These private data may be used by criminals to carry out some malicious activities, resulting in economic or psychological losses to the user. At the same time, it leads to a decrease in the user's trust in the service provider, reduces the use of abandons certain machine learning training platforms or frameworks, and hinders the application and development of models. Therefore, at present, the privacy protection issue is highly valued by scholars and enterprises in the field of machine learning.

Existing approaches to addressing privacy preserving machine learning (PPML) fall into three main categories. Federated Learning (FL) is a distributed machine learning technique that ensures data privacy by performing model training on user devices and then aggregating the updated model parameters to a central server for integration, thus realizing the server's imperceptibility to user data. Split Learning (SL) is a distributed and private privacy-preserving technique that decomposes the training of deep learning models into two phases: local computation and centralized server computation, which enables large-

DOI: 10.54254/2755-2721/69/20241477

scale training of machine learning models while preserving data privacy without the need to transmit the data to a centralized server. Homomorphic Encryption (HE) is an asymmetric cryptography technique that is capable of performing addition or multiplication computations on a ciphertext without decrypting it. This allows the data to remain encrypted while computations are performed, enabling data processing and computation in untrustworthy environments and safeguarding data privacy.

Although homomorphic encryption is computationally inefficient and may make machine learning or deep learning model training time larger, it is still a valuable encryption technique in sensitive scenarios where data privacy needs to be protected. The main objective of this paper is to provide an overview of the state of the art in using homomorphic encryption for machine learning privacy protection and summarize the current technical difficulties, and then discuss the future directions of homomorphic encryption applications in machine learning.

This paper is organized as follows: Section II outlines the history of technology iterations and mainstream algorithms for homomorphic encryption; Section III details homomorphic cryptography-based privacy preservation techniques for machine learning; and Section IV discusses and concludes on the major challenges and future opportunities faced by homomorphic encryption in machine learning privacy preservation.

2. Homomorphic Encryption

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman, the inventors of the RSA encryption algorithm, first introduced the concept of Privacy Homomorphisms [1], describing a method that allows computations to be performed directly on encrypted data without decryption. Twenty years later, Craig Gentry in his doctoral dissertation, first proposed a theoretical Fully Homomorphic Encryption (FHE) scheme, which enables additive homomorphic computation, but the scheme is very inefficient and cannot be applied. In 2009, Gentry first proposed an ideal lattice-based multiplicative homomorphic encryption scheme [2]. This scheme is often referred to as the first-generation fully homomorphic system. In 2010, Gentry [3] described in detail how to realize fully homomorphic encryption and proposed for the first time an integer-based homomorphic encryption scheme DGHV[4], marking a major breakthrough in homomorphic encryption technology. In Gentry's paper, he also proposed a special handling trick for ciphertexts, bootstrapping [5], which makes the noise of a system of finite rank always below a critical value. In 2011, Brakerski and Vaikuntanathan [5] proposed the BGV as a fully homomorphic system based on the LWE assumption. This is a homomorphic encrypted system of finite order that can be transformed into a fully homomorphic system by bootstrapping techniques. It is common to call the BGV system a second generation fully homomorphic cryptosystem. In 2013, Gentry, Sahai and Waters proposed the new GSW fully homomorphic cryptosystem [6]. This system is similar to BGV and is often referred to as a third generation fully homomorphic cryptosystem. Since then, there have been numerous proposals on how to improve the operational efficiency of fully homomorphic systems, but the current overhead problem of homomorphic computation has not yet emerged as a very advantageous solution, either at the hardware level or at the algorithmic level.

Homomorphic encryption can directly calculate the ciphertext without affecting the content of the plaintext, and the result is the same as the result of first calculating the plaintext and then encrypting it. Therefore, homomorphic encryption eliminates the step of decryption and then encryption, and avoids the leakage of user data during the decryption process at the server side, which is also the reason why it is favored in privacy protection.

The two main computational operations are addition and multiplication. It can be assumed that there are plaintexts m1 and m2, belonging to the same plaintext space M. Where E represents the encryption operation and D represents the decryption operation.

HE addition operation:

$$E(m_1) + E(m_2) = E(m_1 + m_2), \forall m_1, m_2 \in M \tag{1}$$

$$D\big(E(m_1) + E(m_2)\big) = D\big(E(m_1 + m_2)\big) = m_1 + m_2, \forall m_1, m_2 \in M \tag{2}$$

DOI: 10.54254/2755-2721/69/20241477

HE multiplication operation:

$$E(m_1) * E(m_2) = E(m_1 * m_2), \forall m_1, m_2 \in M \tag{3}$$

$$D\big(E(m_1) * E(m_2)\big) = D\big(E(m_1 * m_2)\big) = m_1 * m_2, \forall m_1, m_2 \in M \tag{4}$$

Homomorphic encryption can be categorized into three classes based on the number and class of operations that can be supported:

Partially Homomorphic Encryption (PHE): PHE scheme can support additive or multiplicative homomorphism, but not both additive and multiplicative, relatively high efficiency. Common PHE schemes are RSA [7], GM [8], EI-Gamal [9], Paillier [10].

Somewhat Homomorphic Encryption (SHE): SHE supports a finite number of computational operations on the ciphertext, addition and multiplication can occur at the same time, but not an infinite number of computations. Common SHE schemes are BGN[11], CKKS [12].

Fully Homomorphic Encryption (FHE): FHE has the highest level of homomorphic properties and allows arbitrary computations on the ciphertext, including any number and any type of addition and multiplication operations. The emergence of FHE scheme is considered as a major breakthrough in the field of homomorphic encryption. The common FHE schemes are BGV [5], BFV [13].

3. Fully Homomorphic Encryption in PPML

3.1. ML-as-a-service (MLaas)

ML-as-a-service (MLaas) is a platform for outsourcing machine learning application integration to third-party platform providers. As part of cloud computing services, it can provide users (enterprises, universities or individual learners) with a large number of basic machine learning resources, such as cloud servers, APIs, and training data, making it possible for users to use machine learning algorithms without having to build their own services and directly using encapsulated MLaas tools. Due to the increasing demand for AI model application and training in the current market, many Internet companies or cloud server providers have listed MLaas as one of the most important development modules, including Google's Prediction API [14] (soon to be replaced by Cloud Machine Learning Engine), Amazon ML [15], and Amazon ML [16], and Microsoft's Azure ML [13].

The training process and model design of most MLaas platforms on the market are invisible to the user, which means that the user is only selecting the modules they want to use for splicing and parameter modification and uploading private data for machine learning application development or model training when using MLaas. In this process, the user is not able to know the details of the training. The data uploaded by the user is processed and stored on the server side. Obviously, we can't assume that the service provider is able to guarantee data privacy and security in all aspects. In 2024, BMW, a globally recognized car manufacturer, was caught in a cloud storage security crisis. Researcher Can Yoleri reported that during a routine scan, he accidentally discovered that BMW's cloud storage servers (also known as "storage buckets") on Microsoft's Azure platform had been misconfigured and set to public access instead of the expected private state. This serious misconfiguration resulted in BMW's private keys, internal data, and other sensitive information being exposed to the public. This comes on the heels of a similar security issue exposed by Mercedes-Benz, another well-known automaker. Security lab RedHunt reportedly discovered a GitHub private key from a Mercedes employee's code repository, which had access to all of the code on Mercedes' internal corporate GitHub servers. Therefore, service providers should actively explore a series of security measures for the privacy protection of MLaas platforms.

3.2. AI-as-a-service (Alaas)

With the development of AI technology, most cloud service providers are no longer satisfied with implementing a single machine learning algorithm, but have begun to develop AIaas platforms based on AI technologies in multiple fields, such as natural language processing, computer vision, speech recognition, and so on. One of the key advantages of AIaaS is its comprehensiveness. It provides users

DOI: 10.54254/2755-2721/69/20241477

with a unified platform to easily access and integrate various AI capabilities. This includes many of the machine learning services offered by MLaaS, such as model training, evaluation, and deployment, as well as functions in other AI areas, such as sentiment analysis for natural language processing and image recognition for computer vision. Therefore, AIaaS is more suitable for those application scenarios that require a combination of multiple AI technologies, such as intelligent assistants, intelligent customer service, intelligent decision support systems, etc.

3.3. Current Technology

HE can provide a solution. According to the previous description, it is not difficult to think that homomorphic encryption can be used to realize the invisible operation of the cloud server on user data, i.e., the server does not know the private key and does not need to really decrypt the user data, but it directly processes the ciphertext and returns the processed ciphertext to the user, and the user side decrypts it using the private key to get the final result. In this process, first of all, the processor does not decrypt the private data, which avoids some privileged attackers (e.g., company insiders or outsourcing party personnel) who have direct access to the training data and the training process from inserting arbitrary poisoned data into the training set, controlling the data labels, or even directly modifying the training data. Currently, many researchers are actively exploring options to apply HE to MLaas and AIaas.

Adrien et al. [17] proposed a deep neural network based fully homomorphic encryption (FHE) framework TT-TFHE for extending Torus FHE (TFHE) on tabular data and image datasets. It uses a novel family of convolutional neural networks called Truth-Table Neural Networks (TTnet).TT-TFHE provides an easy-to-implement automated toolkit based on Python's open-source Concrete library (CPU-driven, with support for lookup tables) for reasoning about encrypted data. Experimental results show that TT-TFHE outperforms all other FHE setups in terms of time and accuracy. For MNIST and CIFAR-10 image datasets, TT-TFHE also outperforms other TFHE settings by far and competes well with other FHE variants such as BFV or CKKS, while maintaining 128-bit encryption security. In addition, TT-TFHE has a low memory footprint, giving it a big advantage over other FHE frameworks that require tens to hundreds of gigabytes of user memory. And it can be easily scaled to multi-threaded and multi-user server-side.

Phong et al. [18] proposed a new system for deep learning that uses additive homomorphic encryption to protect gradients on cloud servers. While previous CryptoNets [19] relied on pre-trained neural network weights and focused on predicting individual data items, the new system aims to train weights over multiple data sources. The new system utilizes an asynchronous stochastic gradient descent (ASGD) method and additive homomorphic encryption to protect privacy during deep learning, ensuring that no participant information is disclosed on an honest-but-curious server while maintaining accuracy comparable to that of non-privacy-preserving deep learning systems. In contrast, CryptoNets do not involve the training of weights and their performance is limited by sigmoid activation function substitution and computational overhead. The new system also allows learning participants to upload and download parts of the encrypted gradient, adapting to the needs of different scenarios.

Martin et al. [20] have produced a toolset, HE-MAN, that allows homomorphic inference using plaintext models in the Open Neural Network Exchange (ONNX) format to process encrypted input data while preserving the privacy of the model and input data. CryptoNets was the first example of neural network inference using fully homomorphic encryption (FHE), but its performance was limited by the simplification of the activation function and computational overhead. HE-MAN addresses this challenge by supporting homomorphic versions of ONNX canonical operations such as element-level addition, multiplication, matrix multiplication, convolution, average pooling, ReLU activation, and padding, providing a homomorphic runtime time for a subset of ONNX neural networks. Unlike CryptoNets, HE-MAN does not require network-specific optimizations or manual selection of cryptographic parameters, but automatically derives secure and accurate cryptographic parameters. In addition, it supports Concrete and TenSEAL libraries for different usage scenarios. HE-MAN-Concrete utilizes calibration data to optimize ciphertext spacing, while HE-MAN-TenSEAL implements arbitrary linear operations

DOI: 10.54254/2755-2721/69/20241477

via vector-matrix multiplication, sacrificing efficiency for a more flexible model structure. Although HE-MAN does not currently include the ability to train models, it facilitates privacy-preserving machine learning services, allowing model owners to provide inference services without revealing sensitive information. Future directions may include support for additional operations in the ONNX specification for more comprehensive support.

Dongwoo et al. [21] present CONVFHE, a technique for neural network (NN) evaluation on encrypted data, with a particular focus on achieving low-latency single-input inference. By optimizing the way FHE evaluates convolutions and convolutional layers, e.g., using vectorized representations and an improved bootstrap process that packs the inputs into the polynomial coefficients of the ring, a single convolution can be evaluated by a single multiplication without any rotation to achieve homomorphic encrypted inference that is more efficient and suitable for both deep and wide CNNs. In the Extension (Ext) step, the data is extracted and rearranged to fit the bit reversal order. When the input size is much smaller than N, the input can be sparsely packed to reduce the bootstrap time. This method is suitable for mixed-method privacy-preserving machine learning (PPML) inference to reduce the communication cost of convolution. In addition, for CNNs with larger kernels and lower depths, this approach allows for better runtime without sacrificing accuracy.

Kwok-Yan Lam et al. [22] proposed a new model of Hybrid FHE-based PE-NNN that decomposes deep neural networks into a plaintext evaluation part and a ciphertext evaluation part, aiming to solve the privacy problem in AI-as-a-Service. The basic FHE-based PE-NN model limits its use in practical applications due to the high computational cost of executing deep neural networks (DNNs) on encrypted data. To solve the above problem, the model divides the DNN into a plaintext evaluation part and a ciphertext evaluation part. First, the user side executes the open network (OON) and runs it locally in plaintext to extract features and send these features to the server. Then the server evaluates the private network on the encrypted data and returns the encrypted predicted output. Ultimately, only the user with the private key can decrypt and view the results. The advantage of this model is that it can reduce the number of expensive homomorphic encryption evaluations on the server by assigning part of the computational tasks to the user side, and achieve very good system efficiency. At the same time, it guarantees that only the public network is made public, and the user's private network is protected, so that the user can protect the input data and the prediction results by homomorphic encryption, and prevent the cloud server from obtaining sensitive information.

4. Challenges

How to design a fully feasible, fully homomorphic encrypted machine learning privacy preservation system remains a research direction worth exploring. Unlike the federated learning approach, fully homomorphic encrypted machine learning privacy preservation can avoid borrowing a large amount of data for training at the user's end and greatly reduce the burden of users' memory footprint. In this section, a brief list of existing challenges in this area as well as a discussion of potentially viable solutions will be presented.

Performance issues: the biggest bottleneck of homomorphic cryptography is its slow speed. The overhead of FHE is about 10,000 times that of an unencrypted computation. Currently, it can be accelerated in both hardware and software directions. Hardware is mainly based on hardware devices such as GPU\FPGA\ASIC, which is realized by shortening word length, reducing bootstrap operations, lowering memory bandwidth requirements, and expanding GPU microarchitecture. Nowadays, with the help of hardware acceleration technology, the best computing performance under ciphertext can reach 14,173 times that of CPU performance [23]. Software means mainly the optimization of fully homomorphic encryption algorithms. The idea can be the apportionment of bootstrap operations, reducing the overhead of rotary key generation, the use of new data structures and so on. At present, the results of hardware acceleration are relatively better.

Application on RNN: The current application of homomorphic encryption in AI mainly focuses on the combination with CNN, which is because HE is easier to implement on CNN. However, the processing of time series data is also very important, especially with the rapid development of AI video

DOI: 10.54254/2755-2721/69/20241477

generation technology, such as OpenAI's new Vincennes video model SORA. The difficulty is mainly focused on how to deal with the input and output sequences that become longer, visualize the debugging process, reduce the training overhead caused by a large number of encryption and decryption, and so on.

Poor usability of HE: Since problems such as computational overhead have never been well solved, research on homomorphic cryptography has been advancing slowly and there are no good tools to serve AI researchers for direct use. This makes it particularly difficult to study the application of HE in PPML. Researchers need to have both knowledge and an in-depth understanding of computer security and AI to come up with breakthrough solutions. In order to advance the field, it is recommended to establish a good community and repository for HE development.

5. Conclusion

This paper analyzes the feasibility of homomorphic encryption applied in PPML, summarizes the latest research results and techniques, and discusses the existing technical difficulties. Homomorphic encryption plays a crucial role in privacy preserving machine learning (PPML), especially in protecting the privacy of sensitive user data. Although its computational inefficiency may lead to an increase in training time, its unique ability ensures the security of data during processing. Since the concept of privacy homomorphism was introduced in 1978, homomorphic encryption has undergone significant development, ranging from initial theory to full homomorphic encryption schemes for practical applications. This paper analyzes the feasibility of homomorphic encryption applied in PPML, summarizes the latest research results and techniques, and discusses the existing technical difficulties. The current challenge still lies in how to optimize the efficiency of homomorphic encryption for a wider range of practical scenarios. Future opportunities in the field may lie in further algorithmic improvements and hardware optimizations to reduce the overhead of homomorphic computation while enhancing its utility in the field of machine learning. Overall, homomorphic encryption offers promising avenues to address the privacy concerns of PPML, but the technical hurdles that remain to be overcome require continuous efforts from researchers.

References

[1] Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. Foundations of secure computation, 4(11), 169-180.

[2] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing, pp. 169-178.

[3] Gentry, C. (2010). Computing arbitrary functions of encrypted data. Communications of the ACM, 53(3), 97-105.

[4] Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3. Proceedings 29. Springer Berlin Heidelberg, pp. 24-43.

[5] Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). (Leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT), 6(3), 1-36.

[6] Gentry, C., Sahai, A., & Waters, B. (2013). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22. Proceedings, Part I. Springer Berlin Heidelberg, pp. 75-92.

[7] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[8] Goldwasser, S., & Micali, S. (2019). Probabilistic encryption & how to play mental poker keeping secret all partial information. In Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali, pp. 173-201.

[9] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4), 469-472.

DOI: 10.54254/2755-2721/69/20241477

[10] Paillier, P. (1999, April). Public-key cryptosystems based on composite degree residuosity classes. In International conference on the theory and applications of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 223-238.
[11] Boneh, D., Goh, E. J., & Nissim, K. (2005). Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12. Proceedings 2, Springer Berlin Heidelberg, pp. 325-341.
[12] Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, Proceedings, Part I 23, Springer International Publishing, pp. 409-437).
[13] Fan, J., & Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive.
[14] Google Cloud Platform: CLOUD AI. https://cloud.google.com/products/machine-learning. (Accessed: September 2017)
[15] Amazon machine learning. https://aws.amazon.com/machine-learning/. (Accessed: September 2017)
[16] Microsoft Azure: Machine Learning. https://azure.microsoft.com/enus/services/machine-learning. (Accessed: September 2017)
[17] Benamira, A., Guérand, T., Peyrin, T., & Saha, S. (2023). TT-TFHE: a Torus Fully Homomorphic Encryption-Friendly Neural Network Architecture. arXiv preprint arXiv:2302.01584.
[18] 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.
[19] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In International conference on machine learning, PMLR, June, pp. 201-210).
[20] Nocker, M., Drexel, D., Rader, M., Montuoro, A., & Schöttle, P. (2023). HE-MAN–Homomorphically Encrypted MAchine learning with oNnx models. In Proceedings of the 2023 8th International Conference on Machine Learning Technologies, , March, pp. 35-45.
[21] Kim, D., & Guyot, C. (2023). Optimized privacy-preserving cnn inference with fully homomorphic encryption. IEEE Transactions on Information Forensics and Security, 18, pp.2175-2187.
[22] Lam, K. Y., Lu, X., Zhang, L., Wang, X., Wang, H., & Goh, S. Q. (2023). Efficient fhe-based privacy-enhanced neural network for ai-as-a-service. Cryptology ePrint Archive.
[23] Kim, J., Lee, G., Kim, S., Sohn, G., Rhu, M., Kim, J., & Ahn, J. H. (2022, October). Ark: Fully homomorphic encryption accelerator with runtime data generation and inter-operation key reuse. In 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO),IEEE, pp. 1237-1254.