Security and Safety, Vol. 4, 2024012 (2025)
https://doi.org/10.1051/sands/2024012
sands.edpsciences.org

Security and Safety of Data in Cloud Computing

Views

Open Access

Other Fields

Recent advances of privacy-preserving machine learning based on (Fully) Homomorphic Encryption

Cheng Hong*

Ant Group, Beijing 100081, China

Received: 31 July 2024 / Revised: 9 September 2024 / Accepted: 10 September 2024 / Published online: 18 October 2024

Abstract Fully Homomorphic Encryption (FHE), known for its ability to process encrypted data without decryption, is a promising technique for solving privacy concerns in the machine learning era. However, there are many kinds of available FHE schemes and way more FHE-based solutions in the literature, and they are still fast evolving, making it difficult to get a complete view. This article aims to introduce recent representative results of FHE-based privacy-preserving machine learning, helping users understand the pros and cons of different kinds of solutions, and choose an appropriate approach for their needs.

Keywords Homomorphic Encryption, Fully Homomorphic Encryption, Machine learning, Privacy-preserving machine learning

Citation Hong C. Recent advances of privacy-preserving machine learning based on (Fully) Homomorphic Encryption. Security and Safety 2025; **4**: 2024012. https://doi.org/10.1051/sands/2024012

1 Introduction

Fully Homomorphic Encryption (FHE) is a technology that allows data manipulation in the encrypted domain without decryption. The idea of FHE was first introduced by Rivest *et al.* [1] in 1978, but it was not until 2008 that the first FHE scheme [2] was proposed by Gentry. FHE was considered just theoretic and impractical in its early ages, but today many different FHE schemes [3–7] have been proposed, with their efficiency greatly improved.

One of the promising applications of FHE is that it could enable privacy-preserving machine learning (PPML) training or inference on encrypted data, protecting data confidentiality and privacy. There already exist many research works that use FHE to build PPML solutions, some of them even come close to the level of industrial deployments. Given different types of FHE schemes and machine learning tasks, the number of possible combinations is vast, and the area is still fast evolving, thus it's often difficult to answer the question below without enough investigation:

Q: Suppose I want to do privacy-preserving machine learning training/inference of model X on data Y using FHE. Are there any candidate methods for my use case? If yes, which one should I choose? What level of efficiency should I expect?

This article tries to help users answer the question by providing a brief view of the recent research progresses of FHE-based PPML, help them choose research works to follow up, or decide whether the works are mature enough for their needs.

* Corresponding author (email: vince.hc@antgroup.com)

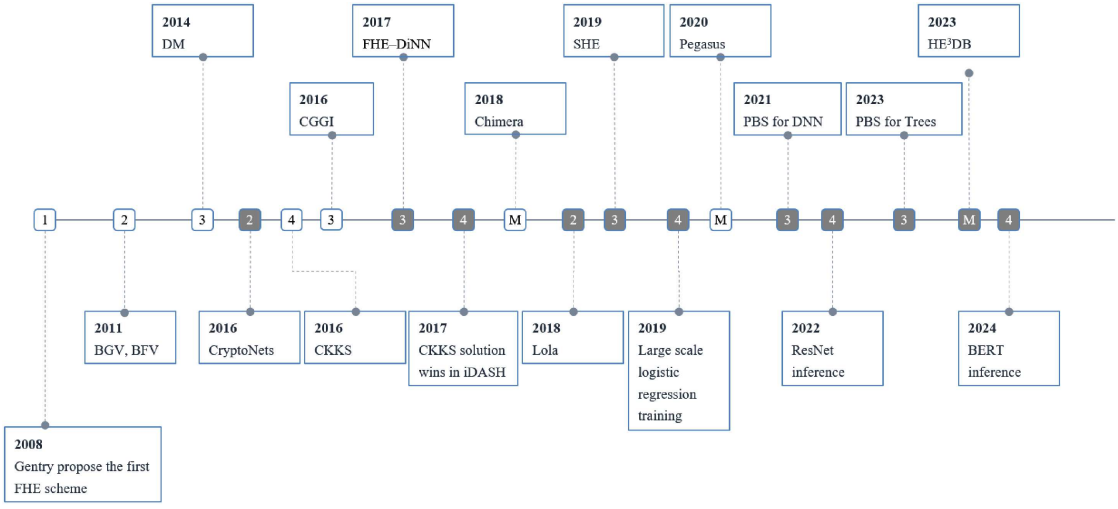

Figure 1. Representative FHE schemes and applications. Numbers in small boxes refer to the generation of FHE, and 'M' refers to mixed-mode FHE. Small boxes with white backgrounds refer to FHE schemes, and dark backgrounds refer to FHE applications

According to a talk by Gentry [8], existing FHE schemes could be divided into four generations as follows:

(1) The first generation FHE refers to Gentry's original design [2] based on ideal lattices, and some other early FHE schemes based on integers such as DGHV [9]. They are too slow to support PPML applications.
(2) The second generation FHE mainly refers to the schemes based on Ring learning-with-errors (RLWE) problem such as BFV [5] and BGV [3]. They can support SIMD (Single Instruction Multiple Data) packing [10], and are already efficient enough for many applications.
(3) The third-generation FHE refers to DM [4] and CGGI [6] features slower noise growth and much faster bootstrapping compared to the second-generation ones. However, they are hard to support SIMD.
(4) The fourth generation FHE is the approximate homomorphic encryption scheme CKKS [7]. It works similarly to second-generation FHEs such as BFV in many aspects, except that it targets approximate arithmetic operations instead of accurate ones, allowing much more efficient homomorphic operations.

We will follow the above classifications, and introduce the applications respectively[1]. A brief overview picture is given in Figure 1. Note that we do not pursue a complete list of related works due to space limits, but (possibly subjectively) pick representative works that reflect the improvement of concrete efficiencies.

2 Second-generation FHE applications

CryptoNets. Microsoft Research's CryptoNets [16] is one of the first works that demonstrated the capability of homomorphic encryption (HE) in the applications of privacy-preserving neural networks. The authors employed the YASHE [17] scheme to encrypt the clients' input images in a pixel-by-pixel way and applied a 5-layer convolutional neural network (CNN) model on top of the encrypted pixels for inference. For the ReLU activation function which is hard to compute in FHE, they replace it with a simpler squaring function. As a result, the full CNN inference just uses a few multiplication depths and does not require any bootstrapping. The cost is that they require the plaintext model to be trained using the same square activation function.

[1] Note that there are many PPML solutions based on MPC [11–14], the difference is that they use interactive protocols between client and server, while FHE is a fully noninteractive procedure run on the server side. This article focuses on the advances in the FHE area and refers to [15] for a survey of PPML works based on MPC.

CryptoNets introduced two innovative optimizations (which, though common in modern FHE applications, were remarkable in 2016): (1) They encoded a plaintext into multiple ciphertexts using the Chinese Remainder Theorem (CRT), enabling support for a 5× larger plaintext space. (2) They employed the SIMD Packing mechanism [10] encrypts pixels of multiple images into one ciphertext, increasing the system's throughput. As a result, CryptoNets was able to process 4096 encrypted MNIST images within 200 seconds, attaining an accuracy of 99%.

Lola. Lola [18] is a follow-up of CryptoNets. Different from CryptoNets which use one ciphertext for each pixel, Lola designs several kinds of message representations in the ciphertext and chooses the best fit representations for each layer of the network. As a result, they are more than 90× faster than CryptoNets and can support larger datasets such as Cifar-10. Note that Lola still requires the square activation function.

NN training. Training neural networks on homomorphically encrypted data poses a significantly greater challenge compared to inference. Nandakumar *et al.* [19] made the initial attempt in this direction: They employed the BGV scheme to encrypt the input images in a bitwise way and packed the same dimension of data from multiple images within a single ciphertext, enabling parallel training on multiple images. The model utilized the Sigmoid activation function and approximated the classification loss using a quadratic function. Notably, they implemented non-linear operations using lookup tables, as lookup tables are more suitable for bitwise-encrypted inputs. However, training a simple three-layer model on a minibatch of 60 images still requires 1.5 days, which is not quite practical.

3 Third-generation FHE applications

Most of the second-generation FHE applications were restricted in the number of homomorphic multiplication depths because they had to avoid the expensive bootstrapping procedure. In contrast, the third-generation FHE schemes such as CGGI [6] are characterized by their ability to fast bootstrapping, and researchers have begun to consider designing privacy-preserving machine learning solutions based on CGGI.

3.1 Early works using CGGI

FHE-DiNN. FHE-DiNN [20] is one of such pioneering works. It introduces Discretized Neural Networks (DiNNs), where both inputs and parameters are represented as integers within a small fixed range so that the computations are friendly for CGGI. FHE-DiNN also optimizes the BlindRotate step of CGGI so that the computational complexity is halved at the cost of extra key size. Leveraging these enhancements, FHE-DiNN could perform homomorphic inference at the rate of one MNIST image per second. The limitation is that they only tested on small single-hidden-layer models containing 30 to 100 neurons.

TAPAS. TAPAS [21], introduced shortly after FHE-DiNN, also employs CGGI for model prediction. TAPAS targets Binary Neural Networks (BNNs), where inputs and weights are binary values. Compared to FHE-DiNN, TAPAS made experiments on larger models such as 5-layer CNNs, enabling higher accuracy. However, they may require hours to complete a single inference.

SHE. SHE [22] also conducted research on secure inference using CGGI. They encrypt the inputs in a bit-wise way and use CGGI to implement homomorphic boolean gates to construct ReLU and max pooling. For the convolution layer which is unfriendly for boolean operations, they quantize the weights into power-of-2 representations, so that the multiplications could be replaced by cheap bit shifts. According to the results in [22], their inference latency is 3−5× higher than Lola [18], but they can achieve better accuracy because of the lossless ReLU activation.

3.2 Works using PBS

The bootstrapping procedure of the CGGI scheme not only reduces the ciphertext noise but also allows the evaluation of a look-up table function at the same time, this is known as Programmable Bootstrapping (PBS). Chillotti *et al.* [23] and Frery *et al.* [24] are two notable works by the Zama team, utilizing the strength of PBS to achieve privacy-preserving machine learning inference.

Neural Network inference. Chillotti *et al.* [23] focuses on privacy-preserving neural network inferences. Unlike [20, 21], this work does not require specific kinds of neural network models, instead, it's able to perform inference on pre-trained models. For efficiency reasons, they first quantize the model parameters to a short ($\leq$8) bit length. Then the homomorphic evaluation of the ReLU activation function could

be performed via several PBSs. Frery *et al.* [23] demonstrated its ability of homomorphic inference on 20-, 50-, and 100-layer neural networks on the MNIST dataset. Experimental results showed that even the 100-layer model could be completed in tens of seconds on an AWS2 cloud server. Very recently, Zama announced an optimization [25] that uses approximated rounding to further reduce the number of required PBS in ReLU, making the inference procedure more than ten times faster than the original [23]. The downside is that their accuracy on MNIST is more than two percent lower than the plaintext model.

Tree-based model inference. Chillotti *et al.* [24] targets tree-based machine learning models such as decision trees, random forests, and gradient-boosting trees. It investigates how to transform tree-based models into operations compatible with homomorphic encryption. Similar to previous work, they quantize the model parameters to a few ($\leq$8) bits, and convert conditional operations into functions that output 0/1 values, allowing for evaluation using PBS. According to the authors' experiments, the inference time for a tree with depth 5 is typically within 5 seconds, and some instances even achieve sub-second performance. In terms of accuracy, the homomorphic model predictions closely match the plaintext accuracy.

4 Fourth-generation FHE applications

The CKKS [7] FHE scheme could also benefit from the SIMD properties like the second-generation FHEs but with a much higher efficiency. CKKS's drawback is that it can only provide approximated results, but the machine learning procedure is often insensitive to small errors. As a result, CKKS quickly marked a significant leap forward in the performance of PPML applications.

4.1 Highlights of CKKS in iDASH competitions

The famous iDASH secure genome analysis competition [26] has become a place where the world's best cryptography researchers and engineers showcase the efficiency of their PPML solutions. In the homomorphic encryption track of iDASH2017, a challenge was announced to train a logistic regression model upon encrypted data. Among the participants, Kim *et al.* [27] achieved remarkable results using the CKKS scheme. They used $5-7$ degree polynomials to approximate the Sigmoid function, and finished training on a gene dataset with 1579 samples and 18 features in just 6 minutes, attaining the expected accuracy and claiming first place. In contrast, BFV/BGV-based approaches might take tens or even hundreds of minutes to achieve similar goals.

In the next year, the task of iDASH2018's homomorphic encryption track is calculating p-values on encrypted SNPs (Single Nucleotide Polymorphisms). This essentially demanded training multiple models simultaneously. The two winners [28] and [29] both used the CKKS scheme. This highlights CKKS's immense performance advantage in privacy-preserving machine learning applications.

4.2 More complicated tasks using CKKS

Because of its approximated nature, CKKS allows much more efficient bootstrapping techniques compared to BFV/BGV. Thus it's possible to use CKKS to tackle more challenging privacy-preserving machine-learning tasks that require bootstrapping.

Logistic Regression training. Han *et al.* [30] successfully trained a logistic regression model using CKKS on 422 108 samples, each with 200 features. They approximate the Sigmoid function using 3-degree Taylor polynomials and carefully design the packing of training data so that the bootstrapping procedure can be run in parallel. It took them 1060 minutes to train for 200 iterations.

ResNet Inference. Lee *et al.* [31] successfully performed inference on CKKS encrypted data using models up to the scale of ResNet110. They designed more compact SIMD packing methods that are tailored for image data and employed high-degree (more than 20) polynomials to provide a precise approximation for the ReLU activation. New techniques are also introduced to reduce bootstrapping errors. Their ResNet-110 inference took approximately 13 000 seconds (with 75% of the time spent on bootstrapping) on an image from the Cifar-10 dataset and achieved an accuracy comparable to the plaintext inference.

Transformer Inference. Recently NEXUS [32] is even able to perform CKKS inference for transformer models. They designed several novel primitives to tackle the difficult parts of transformers, including attention, softmax, layernorm, and argmax. The running time of NEXUS on a Bert-base model for 128 input tokens is 1103 seconds.

5 Mixed-mode FHE applications

As has been shown above, each FHE scheme has its strengths and limitations. Second and fourth-generation FHEs could support large-scale machine learning applications, but they are expensive to handle non-polynomial operations such as ReLU. On the other hand, third-generation FHEs offer flexibility in non-polynomial computations but struggle with efficient homomorphic addition and multiplication.

Chimera [33] is the first framework that combines different kinds of homomorphic encryptions. It proposed new intermediate formats to facilitate switching between CGGI, CKKS, and BFV ciphertexts.

Pegasus [34] follows the blueprint of Chimera but significantly improves the performance. Instead of employing intermediate formats like Chimera, Pegasus designed direct conversions between CKKS and FHEW formats. It also utilized PBS to compute non-polynomial functions on large-domain LWE ciphertexts. Pegasus could compute arbitrary functions including division, sigmoid, or comparison in just one or two seconds. The limitation is that they are efficient only if the PBS accuracy is below 8-bits. Pegasus made multiple experiments including K-means and decision trees, demonstrated the ability to support such tasks on small datasets (with thousands of samples) in several minutes.

HE^{3}DB [35] further showcases the application of mixed-mode FHE. Although HE^{3}DB focuses on database applications rather than machine learning, their idea is also leveraging Pegasus-like methods to solve both polynomial and non-polynomial problems. The difference is that they use repeated PBS to increase the accuracy of comparisons and optimize the noise growth in scheme transferring.

6 Hardware accelarations

Since FHE is computation-intensive, a popular topic is leveraging hardware such as GPU [36, 37] or FPGA [38–40] to accelerate FHE computations.

Park *et al.* [41] developed a GPU library supporting RNS-CKKS. With an NVIDIA A100 GPU, they can finish ResNet-20 inference on a Cifar-10 image in 8.5 seconds, which is more than 267× faster than the CPU version [31]. They also considered adopting several algorithm optimizations such as replacing ReLU activations with low-degree $(2-3)$ polynomials by searching the neural network architecture. By combining this optimization, they can further bring down the inference time to 1.4 seconds.

FAB [42] is an FPGA-based accelerator for CKKS. By using an Xilinx Vivado FPGA operating at 300 MHz frequency, they can train a logistic regression model on 11 982 samples with 196 features in only 0.1 seconds. This is 370× faster than the baseline run under CPU.

There also exist several application-specific integrated circuit(ASIC) designs for FHE acceleration, claiming speed boosting from several thousands to tens of thousand times compared to CPU, but they were mostly evaluated on simulators rather than real chips. We refer to the survey by Zhang *et al.* [43] for more details.

7 Conclusion

The efficiency of FHE has been greatly improved compared to the days when FHE was first invented. Today it is already practical to use FHE for PPML on small or moderate tasks. Each "generation" of FHE schemes has its own pros and cons for different kinds of application scenarios, and could not replace each other. However, there is still a long way to go before FHE can efficiently support more complex tasks such as Imagenet classification. Hardware-software co-optimization would be the key point in improving the efficiency of FHE applications in the future.

Conflict of interest

The author declares no conflict of interest.

Data Availability

No data are associated with this article.

Acknowledgements

No acknowledgements.

Funding

No fundings are related to this article.

References

[1] Rivest RL, Adleman L and Dertouzos ML. On data banks and privacy homomorphisms. Found Secure Comput 1978; **4**: 169–180.
[2] Gentry C. A Fully Homomorphic Encryption Scheme. Stanford University, 2009.
[3] Brakerski Z, Gentry C and Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans Comput Theory (TOCT) 2014; **6**: 1–36.
[4] Ducas L and Micciancio D. FHEW: bootstrapping homomorphic encryption in less than a second. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, 2015, 617–640.
[5] Fan J and Vercauteren F. Somewhat Practical Fully Homomorphic Encryption. Cryptology ePrint Archive, 2012.
[6] Chillotti I, Gama N, Georgieva M, et al. TFHE: Fast fully homomorphic encryption over the torus. J Cryptol 2020; **33**: 34–91.
[7] Cheon JH, Kim A, Kim M, et al. 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, 2017, Proceedings, Part I 23. Springer International Publishing, 2017, 409–437.
[8] Gentry C. A Decade (or So) of Fully Homomorphic Encryption. https://eurocrypt.iacr.org/2021/slides/gentry.pdf
[9] Gentry C, Sahai A and Waters B. 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, 2013. Proceedings, Part I. Springer Berlin Heidelberg, 2013, 75–92.
[10] Smart NP and Vercauteren F. Fully homomorphic SIMD operations. Designs Codes Cryptogr 2014; **71**: 57–81.
[11] Huang Z, Lu W, Hong C, et al. Cheetah: Lean and fast secure two-party deep neural network inference. In: 31st USENIX Security Symposium (USENIX Security 22), 2022, 809–826.
[12] Lu W, Huang Z, Zhang Q, et al. Squirrel: A scalable secure two-party computation framework for training gradient boosting decision tree. In: 32nd USENIX Security Symposium (USENIX Security 23), 2023.
[13] Juvekar C, Vaikuntanathan V, Chandrakasan A. GAZELLE: A low latency framework for secure neural network inference. In: 27th USENIX Security Symposium (USENIX Security 18), 2018, 1651–1669.
[14] Rathee D, Rathee M, Kumar N, et al. CrypTFlow2: Practical 2-party secure inference. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, 325–342.
[15] Ng LKL and Chow SSM. SoK: cryptographic neural-network computation. In: 2023 IEEE Symposium on Security and Privacy (SP). IEEE, 2023, 497–514.
[16] Gilad-Bachrach R, Dowlin N, Laine K, et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning. PMLR, 2016, 201–210.
[17] Bos JW, Lauter K, Loftus J, et al. Improved security for a ring-based fully homomorphic encryption scheme. In: Cryptography and Coding: 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17–19, 2013. Proceedings 14. Springer Berlin Heidelberg, 2013, 45–64.
[18] Brutzkus A, Gilad-Bachrach R, Elisha O. Low latency privacy preserving inference. In: International Conference on Machine Learning. PMLR, 2019, 812–821.
[19] Nandakumar K, Ratha N, Pankanti S, et al. Towards deep neural network training on encrypted data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[20] Bourse F, Minelli M, Minihold M, et al. Fast homomorphic evaluation of deep discretized neural networks. In: Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38. Springer International Publishing, 2018, 483–512.
[21] Sanyal A, Kusner M, Gascon A, et al. TAPAS: Tricks to accelerate (encrypted) prediction as a service. In: International Conference on Machine Learning. PMLR, 2018, 4490–4499.
[22] Lou Q and Jiang L. SHE: A fast and accurate deep neural network for encrypted data. arXiv preprint arXiv:1906.00148, 2019.
[23] Chillotti I, Joye M and Paillier P. Programmable bootstrapping enables efficient homomorphic inference of deep neural networks. In: Cyber Security Cryptography and Machine Learning: 5th International Symposium, CSCML 2021, Be'er Sheva, Israel, July 8–9, 2021, Proceedings 5. Springer International Publishing, 2021, 1–19.
[24] Frery J, Stoian A, Bredehoft R, et al. Privacy-Preserving Tree-Based Inference with Fully Homomorphic Encryption. arXiv preprint arXiv:2303.01254, 2023.
[25] https://www.zama.ai/post/making-fhe-faster-for-ml-beating-our-previous-paper-benchmarks-with-concrete-ml
[26] http://www.humangenomeprivacy.org

[27] Kim A, Song Y, Kim M, et al. Logistic regression model training based on the approximate homomorphic encryption. BMC Med Genom 2018; **11**: 23–31.
[28] Kim M, Song Y, Li B, et al. Semi-parallel logistic regression for GWAS on encrypted data. BMC Med Genom 2020; **13**: 1–13.
[29] Blatt M, Gusev A, Polyakov Y, et al. Optimized homomorphic encryption solution for secure genome-wide association studies. BMC Med Genom 2020; **13**: 1–13.
[30] Han K, Hong S, Cheon JH, et al. Logistic regression on homomorphic encrypted data at scale. Proc AAAI Conf Artif Intell 2019; **33**: 9466–9471.
[31] Lee E, Lee JW, Lee J, et al. Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In: International Conference on Machine Learning, PMLR, 2022, 12403–12422.
[32] Zhang J, Liu J, Yang X, et al. Secure Transformer Inference made Non-interactive. Cryptology ePrint Archive, 2024.
[33] Boura C, Gama N, Georgieva M, et al. Chimera: Combining ring-lwe-based fully homomorphic encryption schemes. J. Math. Cryptol 2020; **14**: 316–338.
[34] Lu W, Huang Z, Hong C, et al. PEGASUS: Bridging polynomial and non-polynomial evaluations in homomorphic encryption. In: 2021 IEEE Symposium on Security and Privacy (SP), IEEE, 2021, 1057–1073.
[35] Bian S, Zhang Z, Pan H, et al. HE3DB: An efficient and elastic encrypted database via arithmetic-and-logic fully homomorphic encryption. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, 2930–2944.
[36] Jung W, Kim S, Ahn J H, et al. Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs. In: IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021, 114–148.
[37] Fan S, Wang Z, Xu W, et al. Tensorfhe: Achieving practical computation on encrypted data using GPGPU. In: 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA), IEEE, 2023, 922–934.
[38] Riazi MS, Laine K, Pelton B, et al. HEAX: An architecture for computing on encrypted data. In: Proceedings of the Twenty-fifth International Conference on Architectural Support for Programming Languages and Operating Systems, 2020, 1295–1309.
[39] Van Beirendonck M, D'Anvers JP, Turan F, et al. FPT: A fixed-point accelerator for torus fully homomorphic encryption. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, 741–755.
[40] Ren X, Chen Z, Gu Z, et al. CHAM: A customized homomorphic encryption accelerator for fast matrix-vector product. In: 2023 60th ACM/IEEE Design Automation Conference (DAC), IEEE, 2023, 1–6.
[41] Park J, Kim D, Kim J, et al. Toward practical privacy-preserving convolutional neural networks exploiting fully homomorphic encryption. arXiv preprint arXiv:2310.16530, 2023.
[42] Agrawal R, de Castro L, Yang G, et al. FAB: An FPGA-based accelerator for bootstrappable fully homomorphic encryption. In: 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA), IEEE, 2023, 882–895.
[43] Zhang J, Cheng X, Yang L, et al. SoK: Fully homomorphic encryption accelerators. ACM Comput Surv 2022.

Cheng Hong received his PhD degree from the University of Chinese Academy of Sciences, China, in 2012. He is currently the Director of Cryptography and Privacy Research of Ant Group, China. His research interests include information security and applied cryptography.