Theoretical Framework and Application Exploration of Fully Homomorphic Encryption

Weiyang Li *

Computer Science Faculty, Science and Technology College of NCHU, Jiangxi, 332020, China

* Corresponding Author Email: li18210090480@outlook.com

Abstract. With the rapid development of big data and cloud computing, the issues of data privacy and security have gradually attracted widespread attention worldwide. Fully Homomorphic Encryption (FHE), as one of the key technologies for solving privacy computing problems, can perform addition and multiplication operations in the encrypted state, thereby enabling effective computation while ensuring data privacy. This paper reviews the theoretical framework and application exploration of FHE. Firstly, it comprehensively summarizes the theoretical basis, development history, and key mathematical tools of FHE, and focuses on analyzing the algorithm principles and latest progress of mainstream schemes such as Brakerski-Fan-Vercauteren (BFV), Cheon-Kim-Kim-Song (CKKS), and Torus FHE (TFHE). It explores the application potential of FHE in fields such as machine learning, medical health, and cloud computing, and analyzes the core challenges it faces, including performance bottlenecks, bootstrapping overhead, and multi-user management. Finally, it looks forward to future research directions such as the integration of FHE and AI, and the construction of cross-domain privacy computing platforms, emphasizing the significant importance of FHE in promoting the two-way advancement of data security and privacy protection.

Keywords: Fully Homomorphic Encryption, Development History, Main Algorithms.

1. Introduction

With the rapid development of technologies such as big data, artificial intelligence, and the Internet of Things, data security and privacy protection have become important issues in the information age. Although traditional encryption techniques can ensure data security, they often cannot directly process encrypted data. Therefore, how to complete data computation without decrypting the data has become an important issue for solving privacy computing and protecting data [1].

Homomorphic encryption is an encryption method that allows operations to be performed on ciphertext, and the result remains encrypted [1]. According to the types of operations supported, common homomorphic encryption can be divided into two categories. The first category is partial homomorphic encryption (PHE), which only supports addition or multiplication operations. The second category is FHE, which supports addition and multiplication operations and can perform any form of computation on ciphertext [2].

The implementation of FHE relies on some complex mathematical tools, which mainly include three tools: the Learning with Noise Problem (LWE), which is widely used in the construction of encryption algorithms; the Ring Learning with Noise Problem (RLWE), a variant of the LWE problem, is often used to construct efficient homomorphic encryption schemes; and the lattice basis, the security of homomorphic encryption schemes relies on lattice theory, especially in large number multiplication and addition operations [3].

The development of FHE has gone through several important stages. Initially, Gentry proposed it in 2009, using an ideal lattice to construct a fully homomorphic encryption scheme. Later, multiple scholars proposed optimized schemes based on different mathematical constructions, such as BFV, CKKS, and TFHE.

Homomorphic encryption technology makes it possible to perform computations in the ciphertext state. It can ensure data privacy while still enabling complex computational processes. FHE is the most representative scheme among them, as it not only supports addition operations but also

multiplication operations, solving the problem that computation cannot be performed on ciphertext in traditional encryption methods [2].

This paper aims to review the theoretical framework, classic algorithms, and latest developments of fully homomorphic encryption, analyze the basic processes and performance of mainstream FHE algorithms, explore their applications in privacy protection, and look forward to the current challenges and future research directions.

2. Analysis of Main FHE Algorithms and Schemes

Most fully homomorphic encryption (FHE) schemes are based on lattice cryptography, and their core lies in effectively controlling the growth of noise while supporting arbitrary computations. Currently, the mainstream FHE schemes mainly include three categories: BGV/BFV, CKKS, and TFHE.

2.1. Brakerski-Gentry-Vaikuntanathan/Brakerski-Fan-Vercauteren Scheme

The BGV and BFV schemes are suitable for scenarios involving precise integer operations, such as in finance and database queries [4]. They support homomorphic addition and multiplication of integers in the plaintext space.

The main process consists of five parts. First is key generation, which generates the public key, private key, and evaluation key (used for re-linearization) [5]. Next is encryption/decryption, where the plaintext is encrypted into ciphertext, and the private key is used to decrypt and restore it after the calculation. Then comes the homomorphic operation, which supports addition and multiplication. Formulas (1) and (2) show their calculation process.

$$E(m_1) + E(m_2) = E(m_1 + m_2) \tag{1}$$

$$E(m_1) \cdot E(m_2) = E(m_1 \cdot m_2) \tag{2}$$

Next comes noise management, which reduces ciphertext noise through modulus switching and extends the computable depth. Finally, there is re-linearization, which reduces the dimension of the ciphertext after multiplication and avoids ciphertext expansion.

The BGV scheme introduced modulus switching technology in 2012, significantly improving efficiency; BFV is its common variant and is widely implemented in libraries such as Microsoft SEAL. In recent years, the research focus has been on reducing the cost of re-linearization and improving parallel efficiency. For example, Turkoglu et al. proposed an accelerated and parallelized GPU implementation of homomorphic encryption operations in 2022, shortening the computation time for various homomorphic operations [6]. Che et al. proposed the multi-key homomorphic encryption (MKHE) scheme in 2023, whose core feature is that it does not require re-linearization operations, which can reduce the RGSW ciphertext overhead, improve computational efficiency and storage performance, and ensure indistinguishability of chosen ciphertext attack (IND-CPA) security [7]. Figure 1 shows the flow of the new ciphertext expansion algorithm and the homomorphic multiplication algorithm.

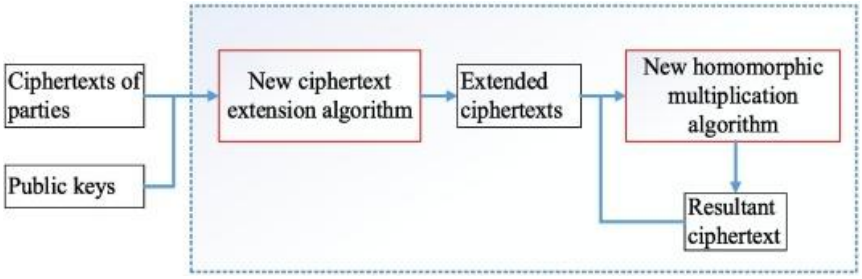

Figure 1. New ciphertext expansion algorithm and homomorphic multiplication algorithm [7]

This scheme is closely centered around eliminating redundant linear operations. Steps 1 (ciphertext construction and introduction of controllable factors) and Step 2 (compact RGSW ciphertext

expansion) lay the foundation for homomorphic computations (especially multiplication), ensuring the controllability of the ciphertext in form and the characteristic of not requiring a conversion key. Step 3 (homomorphic computing algorithm) is the key to achieving non-redundant linearized homomorphic operations. By combining connection and bit decomposition techniques, it ingeniously processes the ciphertext, allowing the multiplication result to be directly used for subsequent operations. Step 4 (decryption) is the conclusion of the entire homomorphic encryption process, verifying the correctness of the scheme.

2.2. Cheon-Kim-Kim-Song Scheme

CKKS is currently the only FHE scheme that supports approximate floating-point arithmetic [8]. It is suitable for applications in machine learning inference, statistical analysis, etc. that require handling real numbers [9].

Its main process consists of five parts. The first is key generation and encryption, similar to BFV, and a scaling factor (scaling factor) is introduced in the encoding stage to control precision. The second is a homomorphic operation, which supports addition and approximate multiplication.

A rescaling operation is performed after multiplication. Then, noise management is carried out. Each multiplication causes the scaling factor to be squared and the noise to be amplified. Through rescaling, the noise is reduced while maintaining precision. After that, relinearization and bootstrapping are supported, which allow controlling the size of the ciphertext and refreshing the noise through bootstrapping to support deeper operations. Finally, decryption is performed, and the approximate plaintext value is recovered after decryption and decoding.

In 2024, Naresh et al. applied CKKS to the PPDNN-CRP framework, achieving privacy protection in the application of deep learning from training to inference. It is a comprehensive solution that can effectively protect data privacy while maintaining high performance and resistance to various security threats [8]. Figure 2 shows the privacy protection credit risk analysis framework.

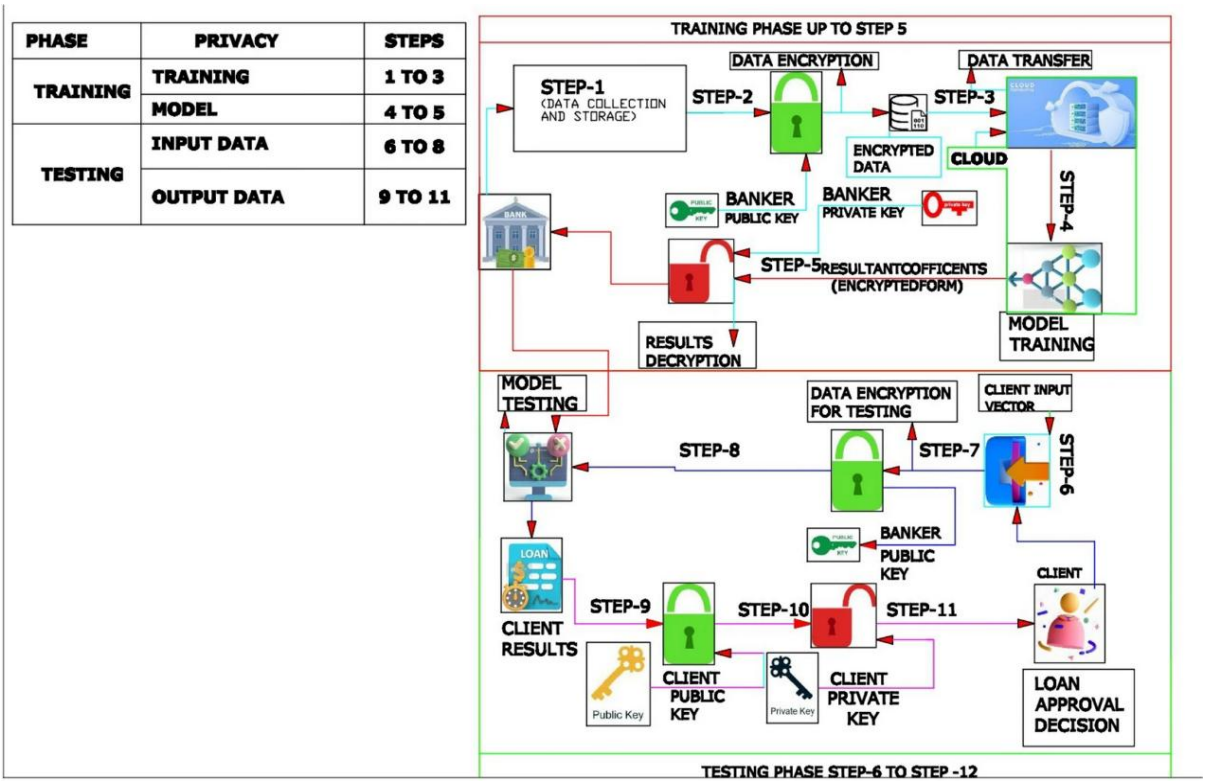

Figure 2. Privacy Protection Credit Risk Analysis Framework [8]

It mainly emphasizes the protection of data, models and outputs during the training and inference processes: Firstly, integrate deep neural networks (DNNs) with homomorphic encryption (HE). Secondly, in the privacy protection training stage (steps 1-5), the PPDNN-CRP framework ensures

that the original sensitive information will not be leaked during the model learning process of the data. At the privacy protection inference stage (steps 6-10), the PPDNN-CRP framework will process the user-submitted credit information, the prediction model, and the final prediction results in an encrypted state to prevent the leakage of sensitive information. Then comes the performance evaluation, where this scheme evaluates PPDNN-CRP on real-world datasets and compares it with other models to verify its performance. Finally, the security analysis is conducted to confirm that it can effectively mitigate data poisoning, avoid attacks, member inference, model inversion, and model extraction, and other threats.

2.3. Torus Fully Homomorphic Encryption Scheme

TFHE is suitable for Boolean circuit operations and is renowned for its efficient gate-level self-bootstrapping (Gate Bootstrapping), which can perform any logical operations on bit data [10].

Its main process consists of five parts. The first is key generation, which includes TLWE/TRLWE keys, bootstrapping keys, and key switching keys. The second is encryption/decryption, where the bit plaintext is encrypted into TLWE/TRLWE ciphertext. Then come the homomorphic logical operations, which support Boolean gate operations such as AND, OR, and XOR, all of which refresh the noise through self-bootstrapping and keep the ciphertext usable. After that, there is the lookup table (LUT), which uses preset tables to achieve efficient computation of any Boolean function, suitable for building complex logical functions. Finally, there is noise management, where a fast self-bootstrapping operation is used to refresh the noise after each gate operation, usually completed in milliseconds.

TFHE was initially proposed by Chillotti et al. in 2017. FHEW, as its predecessor, supports initial bit-level FHE [9]. Benamira et al. proposed TT-TFHE in 2023, which combines the lookup table capability of TFHE and neural networks, enabling inference on encrypted image data [11]. Petrean et al. developed the FHE_string_search algorithm in 2024, performing string search in ciphertext and applying it to Yara rule evaluation [12].

It mainly achieves the goal of efficiently conducting malware detection under privacy protection. Firstly, the server encrypts the Yara pattern: encrypting the Yara rule pattern to ensure data privacy. Secondly, the client performs homomorphic matching, using the TFHE-based FHE_string_search algorithm in the local encrypted file state for matching without decrypting the file. Then, the client evaluates the rule conditions: the matching results remain encrypted, and the client can directly evaluate the Yara rule conditions, avoiding additional communication with the server, further enhancing privacy and efficiency. Finally, the server decrypts the final result, only the final matching results are decrypted, while the original sensitive data is always protected.

3. Current Limitations and Future Prospects

The application prospects of FHE are vast, especially in the fields of data privacy protection and privacy computing. In the field of encrypted inference and machine learning, through the training and inference process of encrypted models, data privacy can be protected [1]. In the field of encrypted processing of medical data, using FHE to encrypt medical data enables privacy-protected data analysis [3]. In the fields of cloud computing and edge computing, FHE can perform privacy-protected computing in a cloud environment, achieving encrypted computing [3].

Although FHE technology has made significant progress, it still faces the following challenges. Firstly, there is a performance bottleneck. The computational complexity of the algorithm is high, and the size of the ciphertext and the computing delay greatly limit the practicality of FHE [2]. Secondly, the cost of Bootstrapping, the re-encryption process in the computation cost is high, and it needs further optimization [1]. There are also key management and multi-user systems. In a multi-user environment, effectively managing keys and ensuring privacy remains a challenge [2].

The future development trends of FHE includes the following aspects. Integration with AI, the training and inference of encrypted large models become an important research direction for FHE [1].

Construction of cross-domain privacy computing platforms, establishing cross-domain privacy computing platforms to achieve more extensive applications [3]. Policies and compliance, with the formulation of privacy protection policies, FHE will play a greater role in data protection [2].

4. Conclusion

In the context of the continuous development of big data and cloud computing, fully homomorphic encryption (FHE) is gradually becoming an important technical support for data privacy protection and secure computing. Through a systematic review of the theoretical framework and development history of FHE, it is clearly observable that this field has continuously evolved from the initial conception, theoretical breakthroughs, to engineering implementation. Notably, mainstream solutions such as BFV, CKKS, and TFHE not only provide diverse implementation paths in terms of algorithm design and mathematical tools, but also offer targeted solutions for security computing requirements in different application scenarios. At the application level, FHE has demonstrated significant potential in areas such as medical health data analysis, financial data processing, cloud intelligent services, and cross-institutional collaborative computing. However, it is necessary to acknowledge that FHE still faces significant bottlenecks in terms of performance overhead, bootstrapping efficiency, and key management in multi-user environments. These challenges restrict its promotion and popularization in large-scale production environments. With the deep integration of artificial intelligence and privacy computing, the research on FHE not only requires continuous breakthroughs in algorithm optimization and hardware acceleration, but also should focus on the construction and standardization of cross-domain platforms, thereby forming scalable and implementable solutions. In the future, with the deepening of related research and the drive of industrialization demands, FHE is expected to play a core role in achieving the goal of making data available but not visible, and lay a solid foundation for building a trustworthy and secure digital society. Therefore, fully homomorphic encryption is not only a key link in the privacy protection technology system, but also an important engine driving the coordinated development of data security and intelligent computing.

References

[1] Bai Lifang, Zhu Yuefei, Li Yongjun, Wang Shuai, Yang Xiaoqi. Research progress of fully homomorphic encryption. Journal of Computer Research and Development, 2024, 61 (12): 3069 - 3087.

[2] Dai Yiran, Zhang Jiang, Xiang Binwu, Deng Yi. Overview on the research status and development route of fully homomorphic encryption technology. Journal of Electronics & Information Technology, 2024, 46 (5): 1774 - 1789.

[3] Marcolla C, Sucasas V, Manzano M, et al. Survey on fully homomorphic encryption, theory, and applications. Proceedings of the IEEE, 2022, 110 (10): 1572 - 1609.

[4] Zhang X, Xu C, Jin C, et al. Efficient fully homomorphic encryption from RLWE with an extension to a threshold encryption scheme. Future Generation Computer Systems, 2014, 36: 180 - 186.

[5] Dilip G. RETRACTED ARTICLE: An efficient privacy preserving on high-order heterogeneous data using fuzzy K-prototype clustering. Journal of Ambient Intelligence and Humanized Computing, 2021, 12 (5): 5191 - 5203.

[6] Türkoğlu ER, Özcan AŞ, Ayduman C, et al. An accelerated GPU library for homomorphic encryption operations of BFV scheme. 2022 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2022: 1155 - 1159.

[7] Che X, Liu L, Wang B, et al. multi-key homomorphic encryption with tightened RGSW ciphertexts without relinearization for ciphertexts product. Journal of King Saud University-Computer and Information Sciences, 2023, 35 (10): 101794.

[8] Naresh VS, Ayyappa D. PPDNN-CRP: CKKS-FHE enabled privacy-preserving deep neural network processing for credit risk prediction. Computational Economics, 2024: 1 - 25.

[9] Chillotti I, Gama N, Georgieva M, et al. TFHE: fast fully homomorphic encryption over the torus. Journal of Cryptology, 2020, 33 (1): 34 - 91.

[10] Boura C, Gama N, Georgieva M, et al. Chimera: combining ring-LWE-based fully homomorphic encryption schemes. Journal of Mathematical Cryptology, 2020, 14 (1): 316 - 338.

[11] Benamira A, Guérand T, Peyrin T, et al. TT-TFHE: a torus fully homomorphic encryption-friendly neural network architecture. arXiv preprint arXiv: 2302.01584, 2023.

[12] Petrean DE, Potolea R. Homomorphic encrypted Yara rules evaluation. Journal of Information Security and Applications, 2024, 82: 103738.