Proceedings of the 3rd International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/135/2025.20959

Advances and Applications in Fully Homomorphic Encryption Research

Dikai Zhao[1,a,*]

[1]*College of Computer Science, Chongqing University, Shapingba District, Chongqing, China*
a. 20221252@stu.cqu.edu.cn
**corresponding author*

Abstract: With the rapid advancement of information technology and the widespread adoption of cloud computing, data security and privacy protection have increasingly become global priorities. In this context, Fully Homomorphic Encryption (FHE) has emerged as a sophisticated encryption technology capable of performing arbitrary computations on encrypted data without the need for decryption, thereby attracting significant interest from both academia and industry. Initially proposed by Rivest et al. in 1978 and practically realized by Gentry in 2009, FHE has evolved through four generations of schemes, each introducing novel construction methods and optimization techniques to enhance security and computational efficiency. Central to modern FHE schemes are lattice-based hard problems such as Learning with Errors (LWE) and Ring-Learning with Errors (RLWE), which provide robust resistance against quantum computing attacks. Additionally, advancements in optimizing the bootstrapping process and exploring hierarchical structures have further improved the practicality and performance of FHE. FHE applications span diverse fields, including cloud computing, artificial intelligence, and blockchain technology, demonstrating its immense potential in ensuring data privacy and facilitating secure computations. However, FHE still faces significant challenges related to computational efficiency, implementation complexity, and application scalability. Future research directions aim to enhance computational performance, broaden application scenarios, strengthen security measures, simplify implementation processes, and develop multi-modal and hybrid encryption schemes. Through a comprehensive review of FHE's development, current progress, applications, and challenges, this paper seeks to provide researchers and engineers with a thorough understanding of the FHE landscape, thereby promoting its continued advancement and practical utilization.

Keywords: fully homomorphic encryption, data privacy protection, cloud computing, artificial intelligence, quantum security.

1. Introduction

With the swift progression of information technology and the extensive implementation of cloud computing, ensuring data security and safeguarding privacy have emerged as paramount global concerns. In this landscape, fully homomorphic encryption (FHE) has garnered significant research and application interest due to its unique capability to perform arbitrary computations directly on encrypted data. This means that operations can be carried out without the need to decrypt the data

DOI: 10.54254/2755-2721/135/2025.20959

first, and the decrypted outcome will be identical to the result of the same operations performed on the original plaintext. Such functionality provides a strong foundation for protecting data privacy in scenarios like data outsourcing and cloud-based services.

The idea of fully homomorphic encryption was initially introduced by Rivest and his colleagues in 1978, earning it the reputation of being the "holy grail" in cryptography [1]. Despite its potential, the practical application of FHE was hindered for many years by its considerable computational demands. This changed in 2009 when Craig Gentry developed the first practical FHE scheme using ideal lattice constructions [2]. Gentry's groundbreaking work not only demonstrated a viable method for achieving FHE but also introduced the "bootstrapping" technique. This innovation effectively reduces ciphertext noise, allowing for unlimited homomorphic operations without compromising the integrity of the encrypted data.

In recent times, there have been notable advancements in both the efficiency and security of fully homomorphic encryption, particularly in enhancing its resistance to attacks from quantum computers. Lattice-based problems, such as the Learning with Errors (LWE) and its ring-based variant Ring-Learning with Errors (RLWE), have become the cornerstone of modern FHE schemes. Furthermore, researchers have made strides in improving the practicality and performance of FHE by refining the bootstrapping process and exploring hierarchical structures within FHE. To illustrate the evolution and key trends in this domain, several noteworthy research efforts can be highlighted. In 2011, Brakerski and Vaikuntanathan introduced FHE schemes grounded in the RLWE and standard LWE problem assumptions [3,4], marking the transition to the second generation of FHE development. The introduction of the GSW scheme by Gentry et al. [5] initiated the third generation of FHE schemes. The GSW approach utilized the approximate eigenvector method for homomorphic operations, thereby eliminating the need for key switching and modulus switching techniques. Moving forward, the fourth generation of FHE schemes is exemplified by the CKKS scheme proposed by Cheon and his team [6]. The CKKS scheme is tailored to support approximate arithmetic operations within the real number domain, enabling efficient homomorphic addition and multiplication by embedding the message space into the complex hyperplane and leveraging complex number properties.

This article aims to provide a comprehensive overview of the evolution of fully homomorphic encryption, examining its current research advancements and potential future applications, particularly in areas such as cloud computing and machine learning. By comparing and analyzing several prominent FHE schemes, the article will identify the existing challenges faced by this field and explore possible directions for future development and enhancements.

2. Overview of Relevant Technologies

2.1. Lattice Definitions and Difficult Problems on Lattices

2.1.1. Lattice Definitions

In mathematics, a lattice refers to a set of points generated by a finite number of linearly independent basis vectors in the Euclidean space R^n. Specifically, given a set of basis vectors $B = \{b_1, b_2, \dots, b_d\}$ (where $b_i \in R^n$), a lattice $L(B)$ is defined as the collection of all integer linear combinations of these basis vectors:

$$L(B) = \left\{ \sum_{i=1}^{d} a_i b_i \mid a_i \in Z \right\} \tag{1}$$

Here, d represents the dimension of the lattice. A lattice is a discrete subset characterized by periodicity and symmetry, and it is widely applied in fields such as number theory, algorithmic research, and cryptography.

DOI: 10.54254/2755-2721/135/2025.20959

2.1.2. Difficult Problems on Lattices

The Shortest Vector Problem (SVP) seeks to find the non-zero vector of minimal length within a given lattice L. Specifically, let $\lambda_1(L)$ denote the length of the shortest non-zero vector in L (the first successive minimum). The SVP can be formulated as: given L, find a vector $v \in L$ that $\left||v|\right| = \lambda_1(L)$.

The recent vector problem involves finding the vector in a lattice that is closest to a given target point $t \in R^n$. The Closest Vector Problem (CVP) can be described as follows: given a lattice L and a target point t, determine the vector $v \in L$ that minimizes $||v - t||$, which is to find the lattice point that best approximates the target.

The LWE (Learning with Errors) problem involves, given an integer matrix $A \in \mathbb{Z}_q^{m \times n}$ and a vector $b \in \mathbb{Z}_q^m$, along with the equation $b = As + e \ mod \ q$, finding the secret vector $s \in \mathbb{Z}_q^n$, where e is a small noise vector following a certain error distribution. LWE is crucial for constructing lattice-based security systems because it is considered computationally challenging.

The RLWE problem is an extension of the LWE problem over polynomial rings. Given a ring R of dimension n and an error distribution χ, the RLWE problem requires distinguishing whether given samples are drawn from the following distributions: (1) Given a secret vector $s \in R_q$, sample $a \leftarrow R_q$ and error $e \leftarrow \chi$, and output $(b = a \cdot s + e \ mod \ q, a)$, or (2) the uniform distribution over R_q^2. This can enhance computational efficiency in both space and time.

These issues on lattices exhibit a high level of security in computational complexity, serving as a critical foundation for constructing quantum-safe cryptographic schemes. They have also become a focal point of cryptographic research in recent years.

2.2. Definition of Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) is an encryption scheme that allows computations to be performed on ciphertexts without the need for decryption. Its fundamental structure consists of four core algorithms: the Key Generation Algorithm (KeyGen), the Encryption Algorithm (Enc), the Decryption Algorithm (Dec), and the Homomorphic Evaluation Algorithm (Eval).

The key generation algorithm (KeyGen) generates a set of keys, including a public key pk, a private key sk and an evaluation key evk, given a security parameter λ. The public key is used for encryption operations, the private key for decryption operations, and the evaluation key is specifically designed to support homomorphic computations.

The encryption algorithm (Enc) utilizes the public key pk to encrypt the plaintext message m, generating the ciphertext c. This process ensures the confidentiality of data during transmission and processing.

The homomorphic evaluation algorithm (Eval) utilizes the evaluation key evk to perform a specified function f on a set of ciphertexts $(ct_1, ct_2, \dots, ct_k)$, producing a new ciphertext ct_f. This algorithm enables direct computation on encrypted data without the need for decryption.

The decryption algorithm (Dec) utilizes the private key sk to decrypt the ciphertext ct_f into the plaintext result $f(m_1, m_2, \dots, m_k)$. If the decryption process fails, it outputs a failure indicator.

Through these algorithms, fully homomorphic encryption schemes enable the ability to perform complex computations on ciphertexts without decryption, ensuring data privacy while supporting a wide range of computational operations.

DOI: 10.54254/2755-2721/135/2025.20959

3. The Evolution of Fully Homomorphic Encryption Schemes

3.1. The First Generation of Fully Homomorphic Encryption Schemes

The first-generation FHE scheme was proposed by Gentry based on ideal lattices [2]. Its main construction process is as follows: First, a SHE scheme is built based on ideal lattices, with its security relying on the Closest Vector Problem (CVP). Next, the decryption circuit is compressed to make the scheme bootstrappable, with its security dependent on the SSSP assumption. Finally, the compressed decryption circuit achieves bootstrapping through homomorphic decryption, thereby realizing full homomorphism.

3.1.1. Optimization of the First-Generation FHE Schemes

Due to issues such as the imperfection of the key generation algorithm, insufficient proof of the security of the SSSP assumption, and low performance, many researchers have subsequently conducted improvement studies. Gentry proposed a new key generation algorithm and enhanced the security of the SSSP assumption through quantum worst-case/average-case reduction [7]. Stehlé and Steinfeld refined Gentry's key generation algorithm, conducted an in-depth analysis of the SSSP assumption, and introduced a probabilistic decryption algorithm to reduce computational complexity [8].

3.1.2. Advantages and Limitations of the First-Generation FHE Schemes

Advantages: (1) Gentry's first-generation FHE scheme is the first to theoretically achieve fully homomorphic encryption, marking a significant advancement in the field of fully homomorphic encryption research. (2) The construction based on ideal lattices provides security guarantees against quantum computing. (3) It enables arbitrary homomorphic operations on encrypted data, laying the foundation for subsequent applications and research.

Limitations: (1) The construction process is complex, making it difficult to comprehend and implement, particularly during key generation and decryption phases. (2) To achieve bootstrapping, it relies on the SSSP security assumption, which has not been thoroughly researched, thereby increasing potential security risks. (3) The ciphertext size and noise growth rate are relatively rapid, which restricts the scheme's practical applicability.

3.2. The Second Generation of Fully Homomorphic Encryption Schemes

The second generation of Fully Homomorphic Encryption (FHE) schemes was first constructed based on the Learning with Errors (LWE) problem assumption. In 2011, Brakerski and Vaikuntanathan introduced FHE schemes based on the Ring Learning with Errors (RLWE) problem and the standard LWE problem assumption. These contributions marked the entry of FHE schemes into the second generation of development, primarily encompassing the following two schemes:

Scheme based on the RLWE problem [3]: This scheme does not require the generation of lattice bases during the key generation process, and its security is quantumly reduced to the worst-case hardness problem on ideal lattices. The subsequent methods for compressing the decryption circuit and implementing bootstrapping in the scheme still follow the approach of Gentry [2].

The scheme based on the standard LWE problem [4]: This scheme employs relinearization technology, with its security reduced to the hardness of the Shortest Vector Problem (SVP) in the worst-case scenario on arbitrary lattices. Prior to this, all lattice-based FHE scheme constructions relied on ideal lattices within various rings. However, this scheme no longer uses Gentry's decryption circuit compression method[2], instead proposing a novel Dimension-Modulus Reduction technique

DOI: 10.54254/2755-2721/135/2025.20959

that enables the scheme to be bootstrapped without introducing additional security assumptions. Additionally, it introduces a Modulus Switching technique to control the expansion of ciphertext noise, and the noise reduction process does not require the computational public key used in the first generation of FHE schemes, achieving simpler and more efficient noise control.

3.2.1. Optimization of the Second-Generation FHE Schemes

The BGV scheme, proposed by Brakerski, Gentry, and Vaikuntanathan [9], is based on the LWE/RLWE problem, with core technologies including key switching and modulus switching. Key switching relies on two sub-algorithms—BitDecomp and Powerof2—which achieve key switching through bit decomposition and power-of-two operations, ensuring that the dimension of the ciphertext does not excessively expand after homomorphic operations. The BFV scheme, introduced by Fan and Vercauteren [10], is based on the RLWE problem assumption. This scheme incorporates the Residue Number Systems (RNS) and the Chinese Remainder Theorem (CRT) representation methods, further optimizing the efficiency of homomorphic operations.

3.2.2. Advantages and Limitations of the Second-Generation FHE Schemes

Advantages: (1) Based on the LWE problem assumption, security is quantum-reduced to the worst-case SVP problem, enhancing resistance against quantum attacks. (2) The second-generation scheme is the first to support Single Instruction Multiple Data (SIMD) operations, allowing batch processing of multiple plaintexts, thereby improving parallel computing capabilities. (3) Key switching and modulus exchange techniques significantly enhance the efficiency of homomorphic operations, reducing the rate of noise growth, making the scheme more efficient in practical applications. (4) It no longer directly relies on complex ideal lattice constructions, simplifying the implementation of key generation and decryption circuits, thereby improving the practicality and comprehensibility of the scheme.

Limitations: (1) The key switching technique relies on two submodules, BitDecomp and Powerof2, leading to an expansion in key size, which increases storage requirements and management complexity. (2) The bootstrapping implementation necessitates that the underlying lattice problem remains hard even when the approximation factor grows super-polynomially, resulting in a stronger security assumption and higher potential risks. (3) The realization of the key switching technique depends on multiple sub-algorithms, adding to the scheme's complexity and implementation difficulty, which may impact overall performance and reliability.

3.3. The Third Generation of Fully Homomorphic Encryption Schemes

The third generation of FHE schemes began with the GSW scheme proposed by Gentry et al. [5]. The GSW scheme introduced a novel approach for performing homomorphic operations by utilizing the approximate eigenvector method, thereby eliminating the need for key switching and modulus switching techniques.

3.3.1. Optimization of the Third-Generation FHE Schemes

Brakerski and Vaikunthanathan demonstrated that the GSW scheme could achieve Fully Homomorphic Encryption (FHE) with shorter parameters and improved the bootstrapping procedure in the GSW scheme by leveraging Barrington's theorem [11,12], enabling homomorphic operations through branching programs. Alperin-Sheriff and Peikert pointed out that the aforementioned method was highly inefficient[13]. They efficiently constructed decryption circuits with smaller depth by transforming the decryption circuit into an arithmetic circuit, resulting in a bootstrapping method with

DOI: 10.54254/2755-2721/135/2025.20959

superior performance and slower error growth. Additionally, they utilized a gadget matrix to create a simpler variant of GSW, further enhancing bootstrapping efficiency. Ducas and Micciancio effectively instantiated the bootstrapping method proposed by Alperin-Sheriff and Peikert in the FHEW scheme, reducing bootstrapping time to under one second [14]. By introducing Programmable Bootstrapping (PBS) technology, the FHEW scheme achieved homomorphic computation of NAND operations on standard LWE ciphertexts during the bootstrapping process, significantly improving bootstrapping efficiency.

3.3.2. Advantages and Limitations of the Third-Generation FHE Schemes

Advantages: (1) The third-generation FHE scheme achieves homomorphic operations through matrix addition and multiplication, avoiding the expansion of ciphertext dimensions and eliminating the need for additional techniques to control dimension growth. (2) By utilizing techniques such as gadget matrices, it effectively manages noise growth within the ciphertext without requiring modulus switching, thereby simplifying the scheme design. (3) The third-generation FHE scheme significantly enhances the efficiency of the bootstrapping process through various optimization methods (AP and GINX bootstrapping), drastically reducing bootstrapping time. (4) It eliminates the need for public key computation during homomorphic operations, streamlining the process and improving overall efficiency.

Limitations: (1) The third-generation FHE schemes currently do not support Single Instruction Multiple Data (SIMD) homomorphic operations, which restricts their application in certain parallel computing scenarios. (2) Although the security assumptions based on (R)LWE remain robust, the third-generation FHE schemes have somewhat weakened their reliance on these security assumptions compared to the second generation, with the approximation factor of the hard problem being only a polynomial in the dimension n. (3) Similar to the second-generation FHE schemes, the third-generation FHE schemes still require bootstrapping to achieve full homomorphism, which to some extent limits their flexibility and broad applicability.

3.4. The Fourth Generation of Fully Homomorphic Encryption Schemes

In the evolution of Fully Homomorphic Encryption (FHE), the fourth-generation schemes represent encryption methods based on approximate computation. These schemes achieve higher computational efficiency compared to previous generations by performing approximate operations in the real or complex number fields. A typical example of the fourth-generation FHE schemes is the CKKS scheme. Similar to the second-generation schemes, the fourth-generation schemes also rely on the security of lattice problems (such as RLWE), but their primary distinction lies in the adoption of approximate arithmetic operations, which significantly enhance computational speed, particularly in application scenarios requiring the processing of floating-point numbers. The CKKS scheme, proposed by Cheon et al. [6], is designed to support approximate arithmetic operations, enabling homomorphic addition and multiplication in the real number field. By embedding the message space into a complex hyperplane and leveraging the properties of complex numbers, the CKKS scheme achieves efficient homomorphic operations.

3.4.1. Optimization of the Fourth-Generation FHE Schemes

Boemer et al. optimized the operational efficiency of scalar encoding and ciphertext-plaintext addition and multiplication through a complex packing-based approach[15]. This method leverages the structural properties of complex numbers to achieve more efficient data representation and processing. The CHIMERA scheme proposed by Boura et al. integrates multiple RLWE-based FHE schemes[16], including CKKS, TFHE, and BFV. By constructing a common plaintext space,

DOI: 10.54254/2755-2721/135/2025.20959

CHIMERA enables efficient switching between different schemes. Specifically, CHIMERA utilizes bootstrapping techniques to allow ciphertexts to be converted between TFHE and BFV, as well as between CKKS and BFV, with BFV serving as an intermediary for conversions between TFHE and CKKS. This innovation significantly expands the application scope of fully homomorphic encryption, enabling it to better adapt to diverse computational needs.

3.4.2. Advantages and Limitations of the Fourth-Generation FHE Schemes

Advantages: (1) The fourth-generation scheme significantly enhances the speed of homomorphic operations through approximate computation, making it particularly suitable for handling floating-point numbers and complex arithmetic operations. (2) The CKKS scheme can embed messages into the complex number domain, supporting efficient floating-point operations in a homomorphic environment, which is applicable to fields such as machine learning and data analysis. (3) Through various optimization techniques, the fourth-generation scheme achieves a notable improvement in computational efficiency while maintaining security.

Limitations: (1) Due to the use of approximate calculations, the fourth-generation scheme introduces errors during computation, which may affect the accuracy of the results, especially in applications requiring high precision, where caution is advised. (2) The CKKS scheme poses a risk of key extraction in certain application scenarios, particularly in situations where partial plaintext results need to be shared. (3) Although bootstrapping techniques enhance the functionality of the scheme, they also increase implementation complexity and computational overhead, especially when dealing with dense keys, where there remains a trade-off between the probability of bootstrapping failure and precision.

4. Applications of Fully Homomorphic Encryption

4.1. Algorithm Library

Helib[17], developed by IBM, is an open-source homomorphic encryption library written in C++. It can be installed and deployed on multiple operating system platforms, including Windows, macOS, Ubuntu, and CentOS. The library relies on the NTL number theory library and the GMP multi-precision arithmetic library at its core, supporting the implementation of both BGV and CKKS homomorphic encryption schemes. Additionally, Helib includes various optimization codes to enhance the efficiency of algorithms, such as Smart-Vercauteren's ciphertext packing technique and Gentry-Halevi-Smart's optimization algorithms. It supports basic operation instructions like "set," "add," "multiply," and "shift." In 2018, IBM released a new version of the Helib library, which optimized the re-linearization algorithm, improving its efficiency by 15 to 75 times.

Microsoft SEAL[18], developed by Microsoft's Cryptography and Privacy Research team, is an open-source homomorphic encryption library written in C++. It is capable of running in various environments and supports three homomorphic encryption schemes: BFV, BGV, and CKKS. When using the SEAL library, users need to understand many specific concepts of homomorphic encryption. The library can perform homomorphic addition and multiplication operations on ciphertexts but does not support operations such as comparison and sorting of ciphertexts.

PALISADE is an open-source project that offers efficient implementations of lattice-based cryptographic constructions and the latest homomorphic encryption schemes. Focused on usability, PALISADE supports the BGV, BFV, CKKS, and FHEW schemes, along with their variants, including corresponding bootstrapping algorithms. Additionally, PALISADE provides a variety of cryptographic schemes such as post-quantum public-key encryption, proxy re-encryption, multi-party computation, threshold homomorphic encryption, identity-based encryption, attribute-based

DOI: 10.54254/2755-2721/135/2025.20959

encryption, and digital signatures. Currently, the primary development team of PALISADE has integrated the project into its successor, the OpenFHE library [19].

4.2. Typical Application Scenarios

4.2.1. The Application of FHE in Artificial Intelligence

The advancement of artificial intelligence relies heavily on vast amounts of data as its "fuel." How to fully unleash the potential of AI while ensuring data security and privacy has become a critical issue. For instance, different industries, departments, or even various business lines within an organization often form data silos due to concerns over data security or privacy. This results in reliance on limited or single data sources for independent training, thereby affecting the construction and optimization of models. The introduction of Fully Homomorphic Encryption (FHE) technology can effectively mitigate this problem. In multi-party machine learning processes, participants can encrypt both foundational data and models, ensuring that data remains in ciphertext form during learning or other processing operations, with the results still encrypted. This FHE-based privacy-preserving machine learning approach enables collaborative training and optimization among multiple parties without exposing the original data. Intel has conducted extensive work in this field. In December 2018, in collaboration with Microsoft, they released the open-source HE-transformer based on the SEAL algorithm library. This tool supports secure operations on sensitive data within AI systems and can be integrated into neural network implementations in open-source frameworks such as Google's TensorFlow and Facebook's PyTorch. It also serves as the homomorphic encryption backend for Intel's neural network compiler, nGraph.

4.2.2. The Application of FHE in Blockchain

In blockchain technology, the execution of smart contracts typically requires the disclosure of all input parameters to allow other nodes in the network to verify the correctness of the contract execution. However, this approach fails to protect the privacy of the contract caller, especially in scenarios involving sensitive information, such as electronic voting, multi-party auctions, and healthcare contracts, where data providers may hesitate to submit their information. By introducing Fully Homomorphic Encryption (FHE) technology, data can be encrypted using FHE algorithm libraries before being uploaded to the blockchain, and the encrypted data is then stored on the blockchain. Subsequently, smart contracts process the data in its encrypted form through homomorphic operations, ensuring that the results of the contract execution remain encrypted[20]. Finally, these encrypted results are returned to the business layer and decrypted using the FHE algorithm library, thereby safeguarding the privacy of contract information and the security of the data. This method not only protects data privacy but also expands the application scope of blockchain smart contracts, enabling them to function effectively in more scenarios that require privacy protection.

5. Challenges and Prospects

5.1. Enhance Computational Efficiency

Current Fully Homomorphic Encryption (FHE) schemes still have significant room for improvement in terms of computational efficiency and resource consumption. Future research should focus on further optimizing homomorphic operation algorithms, reducing computational complexity, and minimizing the time overhead for encryption and decryption. Additionally, hardware acceleration technologies, such as implementations based on GPUs or specialized cryptographic chips, are expected to significantly enhance the practical application performance of FHE.

DOI: 10.54254/2755-2721/135/2025.20959

5.2. Expand Application Scenarios

As FHE technology continues to mature, its application scenarios will gradually expand into more fields. For instance, medical data analysis, financial privacy protection, and IoT device security are all significant potential areas for FHE. Designing more flexible and diverse FHE solutions tailored to the needs of different application scenarios will further promote its widespread practical use.

5.3. Enhance Security

Although lattice-base FHE schemes offer significant advantages in resisting quantum attacks, the security of FHE schemes still requires ongoing attention and enhancement as computational power increases and cryptanalysis techniques evolve. Future research must further refine security proofs, explore more robust security assumptions, and ensure that FHE maintains sufficient protective capabilities against emerging attack methods.

5.4. Simplify the Implementation and Standardize.

Currently, FHE schemes are relatively complex to implement, lacking unified standards and user-friendly development tools, which hinders their widespread adoption in practical applications. In the future, efforts should focus on developing more accessible programming interfaces, standardized protocols, and frameworks to lower the barrier to entry for FHE technology, thereby promoting its broader application and integration across various fields.

5.5. Multimodal and Hybrid Encryption Schemes

In practical applications, a single encryption scheme may struggle to meet the complex and diverse security requirements. Future research could explore multimodal encryption methods, integrating Fully Homomorphic Encryption (FHE) with other encryption technologies such as Partial Homomorphic Encryption and Attribute-Based Encryption. This approach aims to construct more flexible and efficient hybrid encryption schemes, addressing data security needs across various scenarios.

6. Conclusion

Fully FHE, recognized as a cutting-edge advancement in cryptography, has achieved substantial progress in both theoretical research and practical applications in recent years. This article provides a systematic review of FHE's development history, tracing its evolution from the first generation to the fourth generation of FHE schemes. Each generation has introduced significant innovations in construction methods and optimization techniques, enhancing both security and computational efficiency. The initial generations laid the groundwork with foundational concepts, while later generations incorporated lattice-based problems like Learning with Errors (LWE) and Ring-Learning with Errors (RLWE) to bolster resistance against quantum computing attacks. Detailed analyses are presented on the architectural advancements, including bootstrapping methods and noise management strategies, which have addressed previous limitations and improved the practicality of FHE. At the application level, FHE has shown broad potential across various fields such as algorithm libraries, artificial intelligence, and blockchain technology. In algorithm libraries, FHE enables secure computations without exposing sensitive data, enhancing privacy in software applications. In artificial intelligence, FHE facilitates privacy-preserving data processing and model training, which are crucial for handling confidential information. Blockchain applications benefit from FHE by ensuring transaction integrity and confidentiality, thereby strengthening the security framework of decentralized systems. Despite these advancements, FHE faces significant challenges in practical

DOI: 10.54254/2755-2721/135/2025.20959

deployment. Computational efficiency and high resource consumption remain major obstacles that hinder its widespread adoption. Additionally, the complexity of implementing FHE schemes complicates their integration into existing systems. Addressing these challenges requires ongoing research focused on optimizing homomorphic operation algorithms to maintain security while reducing computational complexity. Future directions include exploring parallel processing, hardware acceleration, and more efficient encoding techniques to enhance the performance and scalability of FHE. By providing a comprehensive overview of FHE's evolution, applications, and current challenges, this paper aims to equip researchers and practitioners with the knowledge needed to advance and implement FHE effectively in various domains.

References

[1] *Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. Foundations of Secure Computation, 4(11), 169–179.*

[2] *Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In Proceedings of the 41st Annual ACM Symposium on Theory of Computing (pp. 169–178). ACM.*

[3] *Brakerski, Z., & Vaikuntanathan, V. (2011). Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Advances in Cryptology – CRYPTO 2011 (LNCS 6841, pp. 505–524). Springer.*

[4] *Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient fully homomorphic encryption from (standard) LWE. In Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science (pp. 97–106). IEEE.*

[5] *Gentry, C., Sahai, A., & Waters, B. (2013). Homomorphic encryption from learning with errors: Conceptually simpler, asymptotically faster, attribute-based. In R. Canetti & J. A. Garay (Eds.), Advances in Cryptology – CRYPTO 2013 (pp. 75–92). Springer.*

[6] *Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In T. Takagi & T. Peyrin (Eds.), Advances in Cryptology – ASIACRYPT 2017 (pp. 409–437). Springer.*

[7] *Gentry, C. (2010). Toward basing fully homomorphic encryption on worst-case hardness. In Advances in Cryptology – CRYPTO 2010 (LNCS 6223, pp. 116–137). Springer.*

[8] *Stehlé, D., & Steinfeld, R. (2010). Faster fully homomorphic encryption. In Advances in Cryptology – ASIACRYPT 2010 (LNCS 6477, pp. 377–394). Springer.*

[9] *Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2012). (Leveled) fully homomorphic encryption without bootstrapping. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (pp. 309–325). ACM.*

[10] *Fan, J., & Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Paper 2012/144. Retrieved from https://eprint.iacr.org/2012/144*

[11] *Brakerski, Z., & Vaikuntanathan, V. (2014). Lattice-based FHE as secure as PKE. In Proceedings of the 5th Conference on Innovations in Theoretical Computer Science. ACM.*

[12] *Barrington, D. A. (1989). Bounded-width polynomial-size branching programs recognize exactly those languages in NC1. Journal of Computer and System Sciences, 38(1), 150–164.*

[13] *Alperin-Sheriff, J., & Peikert, C. (2014). Faster bootstrapping with polynomial error. In Advances in Cryptology – CRYPTO 2014 (LNCS 8616, pp. 297–314). Springer.*

[14] *Ducas, L., & Micciancio, D. (2015). FHEW: Bootstrapping homomorphic encryption in less than a second. In Advances in Cryptology – EUROCRYPT 2015 (LNCS 9056, pp. 617–640). Springer.*

[15] *Boemer, F., Costache, A., Cammarota, R., & Wierzynski, C. (2019). NGraph-HE2: A high-throughput framework for neural network inference on encrypted data. In Proceedings of the 7th ACM Workshop on Encrypted Computing Applications and Homomorphic Cryptography (pp. 45–56). ACM.*

[16] *Boura, C., Gama, N., Georgieva, M., & Jetchev, D. (2020). CHIMERA: Combining ring-LWE-based fully homomorphic encryption schemes. Journal of Mathematical Cryptology, 14(1), 316–338.*

[17] *Homenc. (n.d.). HElib [Software]. Retrieved from https://github.com/homenc/HElib*

[18] *Microsoft. (n.d.). Microsoft SEAL [Software]. Retrieved from https://github.com/microsoft/SEAL*

[19] *Openfheorg. (n.d.). Openfhe-development [Software]. Retrieved from https://github.com/openfheorg/openfhe-development*

[20] *Zhong, Y., Jiang, L., Fang, J., et al. (2022). The principle and algorithm of homomorphic cryptography (p. 42). China Machine Press.*