The Investigation of Fully Homomorphic Encryption: Techniques, Applications, and Future Directions

Yiling Xu
Data Science, University of Washington, 1320 NE Campus Pkwy, Seattle, U.S.A.

Keywords: Fully Homomorphic Encryption, CKKS Scheme, Gentry's First Scheme, BGV Scheme, FHE Application.

Abstract: Fully Homomorphic Encryption (FHE) is an encryption method that enables direct computation on encrypted data without revealing its contents. This review offers a concise yet comprehensive overview of FHE, a pivotal cryptographic technology enabling calculations on encrypted data without necessitating decryption. The paper delves into the core concepts and principles underpinning FHE, elucidating the specific methodologies integral to its application. A thorough exploration of various FHE schemes forms a significant part of the study, where each scheme's functionality, advancements, and their broader implications on the field are meticulously examined. The discourse extends to a critical analysis of FHE's practical applications across diverse sectors, assessing its effectiveness and influence within the contemporary technological milieu. By providing an in-depth look at both the theoretical and practical aspects of FHE, this paper aims to highlight its potential and challenges, offering insights into its evolving role in the landscape of secure data processing and privacy-preserving computations.

1 INTRODUCTION

The journey of cryptography through the ages has been marked by continuous evolution, with each advancement bringing with it a new realm of possibilities. Among these, the concept of Fully Homomorphic Encryption (FHE) stands as a monumental achievement, a beacon of potential in the vast sea of data security. This revolutionary idea, first conceived by Rivest et al. in 1978, lay dormant for decades, more a theoretical curiosity than a practical tool (Wikipedia Contributors 2019). It wasn't until 2009, with Craig Gentry's pioneering work, that FHE transitioned from an elusive cryptographic dream to a tangible, albeit complex, reality. This marked a seminal moment in data security, opening new horizons in how sensitive data is handled and processed (Wikipedia Contributors 2019).

Fully Homomorphic Encryption is a cryptographic paradigm unlike any other. In essence, it allows for computations to be performed on data while it remains in an encrypted state, negating the need for decryption during processing. This groundbreaking capability ensures that data is not just protected at rest or in transit, but also while in use, a feat that traditional cryptographic methods have long struggled to achieve (Van Dijk et al. 2010, Armknecht et al. 2015). The implications of this are vast and profound, particularly in an era where data is not merely an asset but the very lifeblood of the technological existence. In the digital world inhabit, where data breaches and cyber threats loom large, FHE offers a beacon of hope - a means to protect the most sensitive of information in ways previously deemed unattainable.

The inception of FHE traces back to a theoretical framework, but it was Gentry's work that illuminated its practical potential. This exploration commences with an in-depth explication of the fundamental tenets of FHE, progressing into a comprehensive examination of the complex mathematical algorithms that constitute the backbone of this intriguing technological advance. It represents an intellectual voyage across a terrain of polynomial algebra and lattice-based cryptographic techniques, a domain where theoretical constructs converge with practical applications, and the esoteric realms of cryptography are rendered comprehensible. This exploration traverses a landscape of polynomial algebra and lattice-based encryption, marrying theoretical concepts with practical applications (Joye 2021).

Furthermore, this exploration is not confined to the theoretical realm. The practical applications of FHE are as diverse as they are impactful. In the realm

Xu, Y.
The Investigation of Fully Homomorphic Encryption: Techniques, Applications, and Future Directions.
DOI: 10.5220/0012819200004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 253-259
ISBN: 978-989-758-690-3

of cloud computing, FHE offers a transformative approach to data processing, enabling secure computations on encrypted data within cloud environments, thus ensuring privacy and confidentiality. In sectors such as healthcare and finance, where the sensitivity of data is paramount, FHE provides a means to process and analyze personal and financial information without compromising security. Governmental agencies, grappling with the dual challenges of data utility and privacy, can leverage FHE to secure national interests while safeguarding individual rights.

The Defense Advanced Research Projects Agency (DARPA) has initiated a program called Data Protection in Virtual Environments (DPRIVE) (Uppal 2024). This program aims to develop an accelerator for FHE. A key aspect of this initiative is a collaboration between Intel and Microsoft. Intel has signed an agreement with DARPA to participate in the DPRIVE program, focusing on designing an application-specific integrated circuit (ASIC) accelerator to enhance the performance of FHE computations. Microsoft is a key partner in this initiative, responsible for leading the commercial adoption of the technology. The goal is to enable computations on fully encrypted data without needing decryption, significantly reducing potential cyber threats. This program is expected to have a broad impact, including applications in healthcare, insurance, and finance, as it allows organizations to utilize and extract value from sensitive data without risking exposure (Uppal 2024, Darpa.mil 2024, Intel 2024).

The DARPA DPRIVE program is a multiyear effort that will involve several phases, starting with the design, development, and verification of foundational IP blocks, leading to the integration into a system-on-chip and a full software stack (Uppal 2024). The collaboration between Intel and Microsoft is aimed at bringing this technology into commercial use, potentially revolutionizing the way sensitive data is handled across various industries.

Inpher, a technology company, has developed XOR Secret Computing®, which utilizes cryptographic techniques including FHE to enable secure, privacy-preserving analytics and machine learning on encrypted data (Anon 2024). This technology allows organizations to compute and derive insights from encrypted datasets across different jurisdictions and environments without exposing the underlying data. It's particularly beneficial for multi-party collaborations where data privacy is crucial.

In this scholarly examination, it provides a concise overview of the fundamental concepts and underlying principles of FHE. It delves into the specific methodologies employed in the application of FHE, as well as an analysis of the diverse sectors where FHE finds utility. The study encompasses a comprehensive exploration of various FHE schemes, detailing their functionalities, advancements, and the consequent implications they have on the field. Furthermore, it engages in a critical discourse on the contemporary applications of FHE, assessing its practicality and impact in the current technological landscape.

2 METHOD

2.1 The Structure of FHE

Fully Homomorphic Encryption is a type of encryption that allows computation on ciphertexts, generating an encrypted result that, when decrypted, matches the result of operations performed on the plaintext. This means that data can be encrypted and processed without ever exposing it in its unencrypted form, thereby maintaining confidentiality and security.

The structure and workflow of FHE shown in Fig. 1 can be divided into three stages. The first is the encryption stage, which starts with the encryption of plaintext data. Each piece of data is converted into ciphertext using a public encryption key. This encrypted data is now secure and can be stored or transmitted without being read by unauthorized parties (Marget 2022). The second is the calculation stage. The difference between FHE and other encryption methods is that it can directly perform calculations on these encrypted data (ciphertext). Operations such as addition, multiplication, and more complicated functions can be performed without decrypting the data. These calculations are performed using algorithms specifically designed for encrypted data. The third is the decryption stage. Once the necessary calculations are completed, the resulting ciphertext can be converted back into readable plaintext. This is implemented using a private decryption key. The result of decryption is the same as when the same calculation is performed on the unencrypted data.

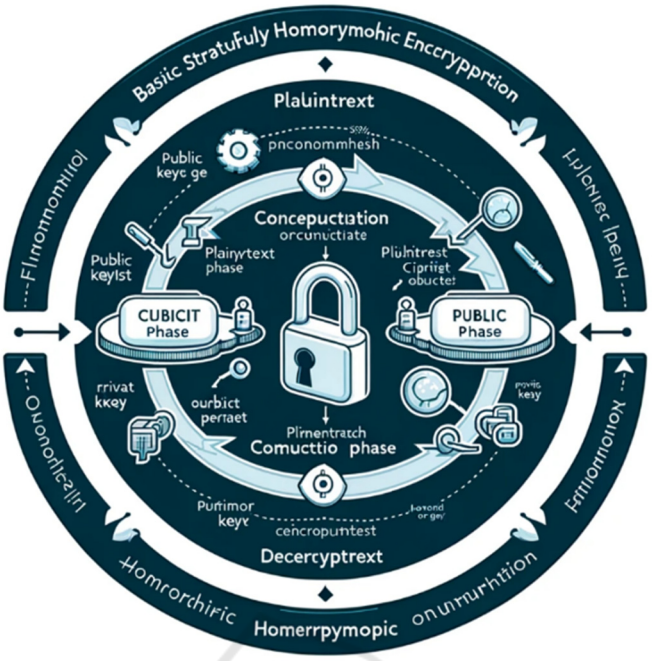

Figure 1: FHE basic structure (Picture credit: Original).

2.2 Key Terminologies

In FHE, several key terminologies are crucial for understanding its framework:

Plaintext: This refers to the original, unencrypted data that is clear and readable.

Ciphertext: Once plaintext is encrypted using FHE, it becomes ciphertext, which is unreadable without the appropriate decryption key.

Encryption Key (Public Key): This is the key used to encrypt plaintext. It's public, meaning it can be shared without compromising security.

Decryption Key (Private Key): This key decrypts the ciphertext back into plaintext and is kept confidential by its owner.

Homomorphic Operations: These are mathematical operations performed on ciphertexts that yield results equivalent to operations done on plaintext.

2.3 Four Properties of FHE

Fully Homomorphic Encryption offers four properties, and here are two pivotal computational abilities on encrypted data: homomorphic addition and homomorphic multiplication. Homomorphic addition, denoted by $\boldsymbol{E(a) \oplus E(b) = E(a + b)}$, allows for the addition of two encrypted values, E(a) and E(b), producing an encrypted sum that, when decrypted, equals the sum of the original plaintext values. This special addition operation works directly on ciphertexts, enabling addition without decryption.

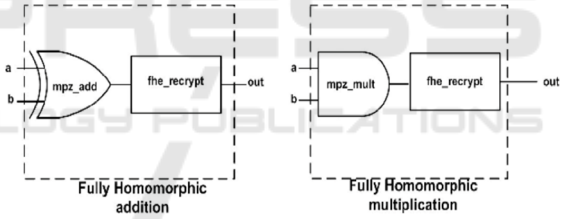

Figure 2: Translating Algorithms to Handle Fully Homomorphic Encrypted Data on the Cloud (Marcolla et al. 2022).

Similarly, homomorphic multiplication, expressed as $\boldsymbol{E(a) \oplus E(b) = E(a + b)}$, permits the multiplication of encrypted data. The multiplication of two encrypted values E(a) and E(b) results in an encrypted product that reveals the product of the plaintext values upon decryption. This is essential as it allows for any computation reducible to additions and multiplications to be executed on encrypted data (Armknecht et al. 2015).

The most significant advantage of FHE lies in its robust data privacy protection. Computation on encrypted data ensures that the underlying data remains confidential throughout the process (Michael 1970). This feature is crucial in scenarios where sensitive data is processed by third parties, such as in cloud computing, or where maintaining data privacy

is legally mandated. FHE enables these third parties to perform necessary computations without ever accessing the actual data, thus preserving privacy.

However, FHE faces a substantial challenge in terms of computational efficiency. The intricate mathematical operations involved in encryption, decryption, and computation on ciphertexts are resource-intensive (Yousuf et al. 2020). This makes FHE considerably slower compared to traditional encryption methods. Despite this, the field is evolving rapidly, with new algorithms and hardware advancements aiming to enhance efficiency. While FHE hasn't reached the efficiency of conventional methods yet, its potential for secure data processing in sensitive applications continues to drive interest and research in the field (Michael 1970).

2.4 Key Schemes in FHE

In the realm of Fully Homomorphic Encryption, several key schemes have been developed over the years, each with unique characteristics and contributions to the field. These schemes represent various approaches to solving the challenges inherent in FHE, such as computational efficiency and noise management. Some of the notable FHE schemes include:

Gentry's First Scheme: Craig Gentry's original FHE scheme, introduced in his groundbreaking 2009 thesis, laid the foundation for FHE. This scheme was based on lattice cryptography and introduced the concept of "bootstrapping" to control the growth of noise in ciphertexts during computations.

BGV Scheme: Named after its creators, the BGV scheme improved upon Gentry's initial work. It offered a more efficient approach to handling noise and enabled more practical implementations of FHE (Khnlil et al. 2017).

Cheon-Kim-Kim-Song (CKKS) Scheme: The CKKS scheme, developed by Jung Hee Cheon et al., is particularly efficient for arithmetic on encrypted floating-point numbers. This makes it well-suited for applications in fields such as data science and machine learning.

Ring Learning with Errors (RLWE) Schemes: These schemes, which are variants of the original Learning with Errors (LWE) problem, use ring-based structures to achieve more efficient encryption and computation. RLWE-based schemes are often more practical for real-world implementations of FHE (Christian et al. 2021, Translating Algorithms to Handle Fully Homomorphic Encrypted Data on the Cloud - Scientific Figure on ResearchGate 2023, Marcolla et al. 2022) as shown in Fig. 2.

Fan-Vercauteren (FV) Scheme: Developed by Junfeng Fan and Frederik Vercauteren, this scheme is notable for its simplicity and efficiency. It has been widely used in various software libraries implementing FHE.

Gentry-Sahai-Waters (GSW) Scheme: This scheme, proposed by Craig Gentry et al. is known for its simplicity and the conceptual clarity it brings to FHE. The GSW scheme has also influenced the development of subsequent FHE constructions.

3 DISCUSSION

3.1 Advancement in FHE

FHE is an encryption technology that allows calculations to be performed on encrypted data without first decrypting it. This means data can remain secure throughout its entire lifecycle – whether at rest, in transit or, more importantly, during processing.

One of the key developments in FHE in recent years has been the introduction of various solutions and libraries that enhance its feasibility and performance. These include CONCRETE, TFHE, FHEW, HEAAN, SEAL, PALISADE, Lattigo and HELib, each supporting different schemes such as CGGI, FHEW, CKKS, BGV and BFV. These libraries and compilers address different aspects of FHE, such as homomorphic arithmetic operations and machine learning workloads (Gorantala et al. 2023). Tools like nGraph-HE, SEALion, CHET, and EVA are designed specifically for machine learning applications, while tools like Cingulata and Encrypt-Everything Everywhere focus on more general computations.

3.1.1 CKKS Scheme

In the field of FHE, one of the most important and widely used schemes is the Cheon-Kim-Kim-Song (CKKS) scheme. The CKKS scheme is particularly notable for its efficiency in handling floating-point calculations, which makes it well suited for applications in machine learning and data analysis. This relevance to machine learning is crucial given the growing importance of artificial intelligence and big data across various industries (Notes on Lattices 2024). The CKKS scheme is able to efficiently manage complex computations while maintaining encryption.

The operation of the CKKS scheme involves several key steps:

Encoding/Decoding: The CKKS scheme encodes real or complex vectors into polynomials and, similarly, decodes these polynomials back to vectors. This procedure uses the roots of cyclotomic polynomials. To decode a polynomial into a vector, the polynomial needs to be evaluated at certain values (specifically the roots of the cycloidal polynomial) (Notes on Lattices 2024). The encoding process, on the other hand, is about finding a polynomial given a vector such that when the polynomial is evaluated at the roots of the cyclotoid polynomial, it is equal to the original vector. This involves solving a system of linear equations using a Vandermond matrix.

Security Basics: The security of the CKKS scheme, like many other FHE schemes, is based on difficult problems in lattice cryptography, specifically the error learning (LWE) and error loop learning (RLWE) problems (Notes on Lattices 2024). These problems are considered difficult for computers, including quantum computers, to effectively solve.

Practical Applications: In practical applications, CKKS can be used to perform complex calculations on encrypted data. For example, in privacy-preserving machine learning, CKKS allows training machine learning models on encrypted data, ensuring the privacy of the data while still enabling valuable computations such as training and inference.

3.1.2 Practical Applications

The practical applications of FHE are very broad, covering various industries such as finance, healthcare, government, cloud computing, security analysis and privacy-preserving machine learning. FHE is also considered quantum-resistant, providing an additional layer of security against potential quantum computing threats (IBM 2021, Iapp 2013).

Healthcare Data Analytics: In the healthcare industry, FHE enables secure and private analysis of sensitive medical data. Hospitals and research institutions can use FHE to perform computations on encrypted patient data for research purposes without exposing personal patient information. The application is particularly valuable for collaborative research involving multiple institutions where data privacy is critical.

Financial Services: In the financial sector, FHE can be used for secure data sharing and analysis. For example, banks can analyze encrypted financial records for fraud detection, risk analysis or credit scoring, while ensuring the confidentiality of individual customer data. This allows for deeper data analysis without compromising privacy.

Cloud Computing and Data Outsourcing: FHE implements secure computing on cloud platforms. Enterprises can store and process encrypted data in the cloud and perform computations on the encrypted data itself. This means that sensitive data such as business analytics, personal information or proprietary algorithms can be processed in the cloud without exposing the actual data to the cloud provider.

3.2 The Challenges of FHE

However, FHE is not without its challenges. The main issue currently hindering its widespread adoption is the complexity of developing efficient FHE applications. Performance, while greatly improved, is still an issue, especially for advanced imperative programs. The restrictive nature of FHE operations and its non-intuitive performance characteristics often require significant transformations of the program to achieve efficiency. To address these issues, recent work has focused on developing compilers like HECO, which aims to transform high-level programs into efficient and safe FHE implementations.

Performance Issues: One of the main limitations of FHE is its computational performance. Initially, the FHE scheme was much slower than operating on unencrypted data, sometimes a trillion times slower. Although progress has been made in this area, and recent advances have reduced performance degradation, FHE operations are still quite slow, up to a million times slower than unencrypted operations (Baffle 2017). This significant speed reduction poses a huge challenge for practical applications, especially in situations where fast data processing is critical, such as database queries.

Implementation Complexity: The development of FHE applications is complex and requires expertise in cryptography and programming. Developers need to choose from various FHE schemes and libraries, such as CONCRETE, TFHE, FHEW, HEAAN, SEAL, PALISADE, Lattigo, and HELib, each with their own advantages and limitations. Furthermore, domain-specific compilers (e.g., ALCHEMY, Marble, RAMPARTS) focus on a subset of computations but suffer from inefficient bootstrapping operations. General-purpose compilers such as Cingulata and Encrypt-Everything-Everywhere offer wider applicability, but are often slower and require specific configuration (Gorantala et al. 2023). The complexity of selecting the right tools and implementing them effectively limits the practical deployment of FHE.

Lack of Unified APIs and Benchmarks: FHE lacks standardized application programming interfaces (APIs) and concrete benchmarks, further complicating its integration with existing systems (Gorantala et al. 2023). The lack of standardization makes it difficult for developers to evaluate the performance and security of their FHE implementations and to ensure compatibility across different systems and applications.

Standardization Efforts: Recognizing these challenges, researchers are continually working to develop FHE standards. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) have launched the FHE standards project, which is currently in the consultation phase. Future phases of standardization may involve the development of standards for application areas including memory encryption, wireless communications and portable devices (Iapp 2013). Key management is an important aspect of FHE and the focus of these standardization efforts, as the current public key infrastructure is not well suited for FHE.

3.3 The Development of FHE in the Future

Regarding future developments, there are ongoing efforts to standardize and benchmark FHE. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) have launched the FHE standards project, which is currently in the consultation phase. This standardization may extend to applications such as memory encryption, wireless communications, and multi-stakeholder software-defined networking (Iapp 2013). Key management is another area of focus, especially given that the current public key infrastructure is not well suited for FHE. In the future they should accelerate FHE performance, standardization and benchmarking, develop user-friendly tools and libraries, collaborative research and development.

4 CONCLUSION

This paper mainly describes the background knowledge of FHE and some FHE methods. This is a cutting-edge encryption technology that can calculate encrypted data without decryption. Great strides have been made in this area, with various schemes such as CKKS, CONCRETE, TFHE and FHEW playing a key role. Despite these advances, FHE still faces challenges such as computational inefficiency, implementation complexity, and lack of standardized APIs and benchmarks. This paper studies in detail the CKKS scheme, which is known for its efficiency in handling real or complex numbers and its suitability for applications such as machine learning. However, an overarching theme in FHE development is balancing its groundbreaking potential for secure data processing with the ongoing challenge of optimizing its performance and usability for wider practical applications. It is anticipated that these challenges and shortcomings can be overcome one by one in the future. This article may lack some actual data, which will be filled in in the future.

REFERENCES

Wikipedia Contributors, Homomorphic Encryption., Wikimedia Foundation, 3 (2019).

M. Van Dijk, C. Gentry, S. Halevi, & V. Vaikuntanathan, Fully Homomorphic Encryption over the Integers 2010.

F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, C. A. Reuter & M. Strand, A Guide to Fully Homomorphic Encryption, EPrint IACR (2015).

M. Joye, Guide to Fully Homomorphic Encryption over the [Discretized] Torus, EPrint IACR (2021) .

R. Uppal, (n.d.). DARPA DPRIVE developing an ASIC for Fully homomorphic encryption (FHE) to ensure Data privacy and Security, International Defense Security & Technology, January 5 (2024).

Darpa.mil, PROgramming Computation on EncryptEd Data (2024).

Intel, Intel to Collaborate with Microsoft on DARPA Program, Intel, January 5 (2024).

Anon, XOR Secret Computing Engine, Inpher, January 5 (2024).

A. Marget, Data Encryption: How It Works & Methods Used, Unitrends, January 27 (2022).

F. Armknecht, C. Boyd, C. Carr, K. Gj.steen, A. J.schke, C. A. Reuter, M. Strand, A guide to fully homomorphic encryption, IACR Cryptol. ePrint Arch., 2015:1192.

G. Michael, Fully homomorphic encryption with applications to privacy-preserving machine learning, Jan (1970).

H. Yousuf, M. Lahzi, S. A. Salloum, K. Shaalan, Systematic review on fully homomorphic encryption scheme and its application, Recent Advances in Intelligent Systems and Smart Applications (2020).

H. Khalil et al. On DGHV and BGV fully homomorphic encryption schemes. 2017 1St cyber security in networking conference (CSNet) (2017).

M. Christian, et al. Multiparty homomorphic encryption from ring-learning-with-errors. Proceedings on Privacy Enhancing Technologies 2021.CONF (2021).

Translating Algorithms to Handle Fully Homomorphic Encrypted Data on the Cloud - Scientific Figure on ResearchGate, 30 Dec (2023).

C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. P. Fitzek and N. Aaraj, Survey on Fully Homomorphic Encryption, Theory, and Applications, in Proceedings of the IEEE, vol. 110, no. 10, pp. 1572-1609, Oct. 2022.
S. Gorantala et al. Communications of the ACM, May (2023).
Notes on Lattices, Homomorphic Encryption, and CKKS. (n.d.). Ar5iv. January 5 (2024).
IBM. Fully Homomorphic Encryption. February 9 (2021)
Iapp. The latest in homomorphic encryption: A game-changer shaping up (2013).
B. Baffle.io. Why Is Homomorphic Encryption Not Ready for Primetime? March 17 (2017).