WEP Warwick Evans Publishing

Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 6
ISET 2024

Progress and Applications of Fully Homomorphic Encryption

Shenglong Zhou

China-Europa College of Engineering and Technology, Shanghai University, Shanghai, China

fengqing@ldy.edu.rs

Abstract. In the era of digital transformation, marked by the widespread adoption of cloud-based technologies such as cloud computing and storage, ensuring data security and protecting user privacy have emerged as critical industry concerns. Fully Homomorphic Encryption (FHE) addresses these issues by allowing various computations to be performed on ciphertexts without the need for decryption, thus safeguarding data security and privacy. This paper explores the mathematical and algorithmic foundations essential for understanding FHE. It provides a detailed analysis of the advancements in FHE schemes over the past decade, focusing on various mathematical problems, including ideal lattices and integers. The discussion extends to three practical applications: cloud computing, machine learning, and electronic voting, highlighting the progress and exploring potential future research avenues and developmental strategies in the field of FHE. This comprehensive review not only underscores the significance of FHE in enhancing data security but also charts a path for its future exploration and integration into emerging technologies.

Keywords: FHE; Lattice; LWE Problem; Bootstrapping; Cloud Computing.

1. Introduction

The concept of homomorphic encryption, initially proposed by Rivest et al. in 1978, centers on creating an encryption mechanism that allows for encrypted information retrieval by exploiting the homomorphic properties of encryption functions. This concept enables computations to be performed directly on ciphertexts without the need for decryption, with the decrypted results matching those that would have been obtained had the operations been performed on the plaintexts. For over three decades, this notion has remained a pivotal focus in the field of cryptography, with the implementation of Fully Homomorphic Encryption (FHE) presenting a pressing challenge for the cryptographic community.

It was not until 2009 that Gentry introduced a groundbreaking FHE scheme, based on ideal lattice design. This scheme, detailed in his seminal paper, constructed a mechanism for partial homomorphic encryption using the bounded encoding problem with a sparse subset sum problem on ideal lattices. It employed the Squash technique to represent the decryption function as a polynomial with a sufficiently low degree, achieving a 'self-lifting' function. This function allows the transformation of ciphertext into an arbitrary function without the decryption key, thereby realizing FHE through what is now known as the Bootstrapping technique.

The progress triggered by Gentry's work ignited widespread interest in FHE. Over the subsequent decade, numerous cryptographers made significant advances in the development of FHE schemes. In 2010, Marten van Dijk and colleagues utilized Gentry's technique to propose a FHE scheme capable of handling integers. By 2011, Zvika Brakerski and Vinod Vaikuntanathan had introduced LWE-based FHE schemes, followed in 2013 by Gentry et al., who proposed FHE schemes based on approximate eigenvectors. In 2017, Cheon et al. developed FHE schemes capable of processing floating point numbers, marking a steady progression towards practical FHE applications.

This survey provides a detailed account of the research dynamics of FHE since its inception, along with its applications, structured as follows: Section II introduces the basic theory, encompassing the mathematical and algorithmic foundations necessary to comprehend FHE schemes. Section III discusses various FHE schemes based on different mathematical problems, analyzing and comparing the security issues they present. Section IV explores the application of FHE in cloud computing,

machine learning, and e-voting. Finally, Section V outlines potential future research directions and development recommendations.

2. Basic Theory

2.1. Mathematical Theory

Polynomial Rings: The mathematical structure, properties of polynomial rings can simplify the computation of encryption and decryption and the complexity of ciphertexts and keys. Let R be a ring, the polynomial P(X) of a single variable X can be defined as the following tableau:

$$P(X) = a_m X^m + a_{m-1} X^{m-1} + \cdots + a_1 X + a_0 \tag{1}$$

Addition of polynomials is achieved by adding the corresponding coefficients, whereas multiplication of polynomials is achieved by expanding and combining like terms, and the operations are tabulated as follows:

$$\sum_{i=0}^{n} a_i X^i + \sum_{i=0}^{n} b_i X^i = \sum_{i=0}^{n} (a_i + b_i) X^i \tag{2}$$

$$(\sum_{i=0}^{n} a_i X^i)(\sum_{j=0}^{m} b_j X^j) = \sum_{k=0}^{m+n} (\sum_{\mu+\nu=k} a_\mu b_\nu) X^k \tag{3}$$

Modulo arithmetic: Modulo arithmetic ensures that the result of arithmetic operations in an encrypted domain remains in that domain, which helps to maintain the validity and consistency of the data. In addition, modulo arithmetic avoids overflow problems in large number operations and improves the efficiency of computation. FHE schemes rely on the homomorphic properties of addition and multiplication, which are only naturally preserved with modulo operations.

Gaussian distribution: the general form of the normal (or Gaussian) distribution χ of a random variable $x \in R$ is represented by the probability density function as:

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4}$$

2.2. Algorithm Theory

Key generation algorithm: the randomness algorithm, $pk, sk \leftarrow \text{KeyGen}\ 1^\lambda$, accepts the security parameter λ as input and produces the secret key sk, public key pk, and (public) evaluation key evk that are required for the homomorphic operation to be performed by the ciphertext.

Encryption algorithm: the randomness algorithm, $c \leftarrow \text{Enc}\ pk,\text{m}$, produces the ciphertext c after receiving the message m in the message space and the public key pk as inputs.

Decryption algorithm: deterministic algorithm, $m \leftarrow \text{Dec sk,c}$,input secret key sk and ciphertext c and output message m. If the decryption algorithm does not succeed in recovering the encrypted message m, then output $\perp$.

Evaluation algorithm: randomness algorithm, $c^* \leftarrow \text{Eval}\ pk, C, c_1, c_2, \ldots, c_t$, input a string of ciphertexts (c1, ..., ct), the function f, and the evaluation key evk. It generates a ciphertext cf that decrypts the outcome of f's evaluation to (m1, ..., mt).

3. Progress in The Study of FHE Schemes

Mainstream FHE systems currently in use are mostly based on two distinct problems: the approximate greatest common divisor (AGCD) difficulty problem on integers and the lattice difficulty problem. Figure 1 depicts the classification of classical FHE schemes. This section first provides a brief analysis of integer-based FHE schemes in Section 3.1, and then presents the construction principles of the four generations of classical FHE schemes based on the lattice-hard problem and their limitations in Sections 3.2-3.5. Finally, a comparative analysis of mainstream FHE schemes is presented in section 3.6.

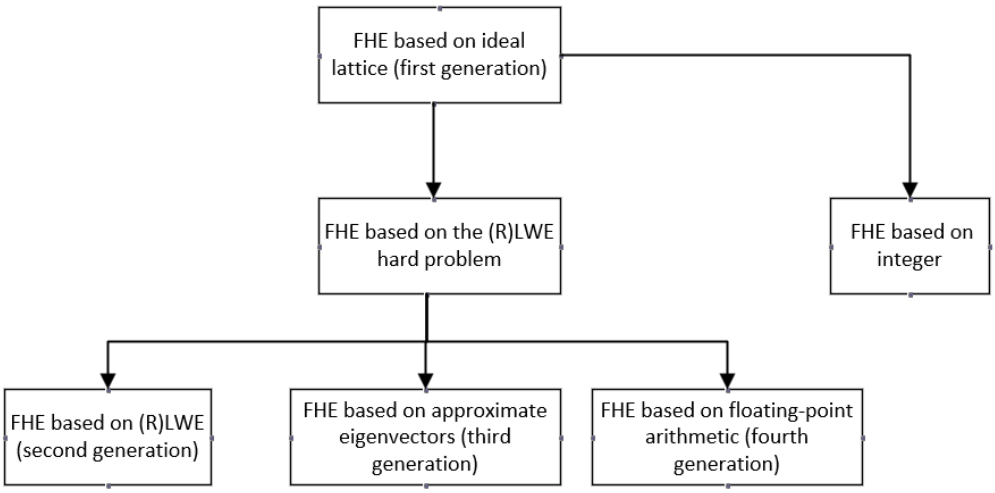

Figure 1. Schematic classification of FHE schemes (Photo credit: Original).

3.1. FHE Scheme Based on Integer

Understanding ideal lattice-based FHE schemes requires more mathematical and algorithmic foundations, so it is necessary to study FHE schemes based on simple algebraic structures (e.g., integers).In 2010, Dijk et al. first proposed an integer-based FHE scheme (i.e., the DGHV scheme), which replaces the complex matrix and vector operations on the lattice with integer modulo operations, which fully follows the Gentry regime construction idea, and its SHE scheme construction relies on the AGCD difficulty assumption. This scheme has the disadvantages of high computational complexity and large public key capacity, so in 2012, Coron et al. further optimised quadratic encryption and reduced the public key size. Using Brakerski's technique, Coron et al. enhanced the DGHV scheme in 2014.Inspired on Brakerski's work, Cheon et al. introduced a reduction from LWE to AGCD in 2015, and then developed a new variant of the AGCD-based FHE scheme, which was constructed to be the first DGHV variant that does not need to rely on the SSSP assumption.

3.2. First-Generation FHE Scheme Based on The Ideal Lattice

The 1st generation ideal lattice-based FHE scheme was proposed by Gentry , and its construction process is as follows: firstly, an ideal lattice-based SHE scheme is constructed; secondly, the complexity of the decryption function of the SHE scheme is reduced by adding some extra information about the key in the evaluation key, which makes the scheme bootstrappable, and the security relies on the sparse subset summation problem (SSSP); finally, the compressed Finally, homomorphic decryption is performed on the compressed decryption function to achieve bootstrapping, thus achieving full homomorphism.

Gentry's FHE scheme achieves a significant progress in the theory of FHE, lays the theoretical foundation of FHE, and provides an important direction for subsequent research. However, it still has many limitations: 1) The construction is relatively complex, making the implementation and understanding of Gentry's scheme require in-depth mathematical knowledge and complex algorithms, and it is difficult to be widely promoted and applied. 2) The noise and ciphertext size are too large,

thus making the bootstrap procedure too inefficient, requiring a large amount of computational resources and time costs, which is not practically relevant. 3) The scheme's bootstrap can be achieved only by reducing the complexity of the decryption function can achieve the bootstrap of the scheme, leading to the introduction of the SSSP assumption that fails to fully verify the security.

As a result, there have been many subsequent studies based on the Gentry regime. The first improved optimisation scheme for the Gentry regime was presented by Smart et al. at PKC 2010. The scheme has smaller sized keys and ciphertexts, uses a master ideal lattice and introduces batch processing techniques. The batch version of the scheme can encrypt plaintext vectors packed into a single ciphertext using the Chinese Residual Theorem (CRT). This technique allows multiple messages to be processed at the same time, enabling parallel execution of re-encryption. In 2011, Halevi et al. further simplified the compression process and improved the bootstrap efficiency.

3.3. Second-Generation FHE Scheme Based on LWE

The 2nd generation FHE schemes are based on the LWE problem assumption to construct FHE schemes. Using bootstrap approaches, Brakerski and Vaikuntanathan introduced FHE schemes based on the RLWE problem and the LWE problem in 2011.The former scheme extends the Gentry regime in reducing the complexity of the decryption function as well as the bootstrap approach, but with a slight difference in the construction of the SHE scheme, where the generation of lattice bases is no longer required for key generation. The latter scheme employs a relinearization technique in constructing the SHE scheme to reduce the size of the multiplicative ciphertext text to achieve key conversion; and the scheme no longer extends the compression method of the Gentry regime, but proposes a new technique, i.e., dimensional-modular reduction, which can control the growth of the ciphertext noise more efficiently, and moreover, it no longer relies on the SSSP assumptions, and the security is is more reliable.

Soon after, Brakerski proposed a hierarchical fully homomorphic scheme which can achieve ciphertext noise reduction without exploiting bootstrap procedures, thus saving significant computational costs [1]. They established the BGV scheme and supplied batch processing and analogue-to-digital conversion tools based on this methodology. The scheme operates as follows:

Key generation: taking the security parameter λ as input, randomly select an encrypted element $s \in \chi$, and set the key $\mathrm{sk} = \mathbf{s} = (1, s) \in R_q^2$, randomly generate $a' \in R_q$ and compute $b = a's + 2e$, where e is the random error in χ. Finally, output sk and $\mathrm{pk} = \mathbf{a} = (b, -a')$.

Encryption: input public key $\mathrm{pk} \in R_q^2$ and message $m \in R_2$, which converts m into vector $\mathrm{m} = (m, 0) \in R_q^2$ and randomly selects $r, e_0, e_1 \in \chi$. Finally, the output ciphertext $\mathrm{c} = \mathrm{m} + 2(e_0, e_1) + \mathrm{a}r$.

Decryption: input key $\mathbf{s} \in R_q^2$ and ciphertext $\mathbf{c} \in R_q^2$, compute $(\mathrm{c,s}) = c_0 + c_1 s = m + 2e_0 + 2e_1 s + 2er$ and output $((m + 2(e_0 + e_1 s + er))modq)mod2 = m.$

Brakerski subsequently proposed an additional BGV scheme variation [2]. This scheme reduces the error increase from homomorphic multiplication from exponential to linear. Fan and Vercauteren built and improved the RLWE version of this technique, which they called the FV scheme [3].

The construction of the 2nd generation FHE scheme still requires the creation of a SHE scheme followed by full homomorphism via bootstrap techniques, but the construction process no longer requires compression-decryption circuits and thus no longer relies on SSSP assumptions, which makes it more secure and efficient compared to the 1st generation.

The limitations of Generation 2 are: 1) The storage cost is large and the key conversion technique increases the size of the key. 2) It is more difficult to Select the relevant parameters, and the parameter selection directly affects the scheme's effectiveness and security, too large a parameter leads to inefficiency, while too small a parameter may affect security.

3.4. Third-Generation FHE Scheme Based on Approximate Eigenvectors

In 2013, Gentry et al. first proposed an approximate eigenvector-based FHE scheme (GSW scheme). The homomorphic operations in the GSW scheme are all simple matrix addition and multiplication operations, which well solves the problem of the ciphertext dimensionality expansion and controls the rate of the noise growth by decomposing the high-paradigm matrices in the noise entries into a low-paradigm binary bit matrix. The working principle of the GSW scheme operates as follows:

Key generation: output key $\mathrm{s} = (1, s_2, \ldots, s_n) \in \mathbb{Z}_q^n$, where s_i' is randomly chosen and $A \in \mathbb{Z}_q^{n \times n}$.

Encryption: compute the ciphertex$C = m\mathrm{I}_n + RA$ where $m \in \mathbb{Z}_q$is the message, the identity matrix is denoted by I_n, while R is an n×n random matrix whose coefficients are in $\mathbb{F}_2$.

Decryption: compute $C\mathrm{s} = m\mathrm{I}_n\mathrm{s} + RA\mathrm{s} = m\mathrm{I}_n + R\mathrm{e} \approx m\mathrm{I}_n s$. Since R is small, if $A\mathrm{s} \approx 0$, then $RA\mathrm{s} \approx 0$. Finally, output the vector $x \approx m\mathrm{I}_n\mathrm{s} \approx (ms_1, \ldots, ms_n)$.

The GSW scheme still needs to achieve full homomorphism with the help of bootstrap procedure, so the FHEW and TFHE schemes were proposed in 2014 and 2016, respectively, to optimize the bootstrap procedure and reduce the bootstrap time to less than 0.1 sec, and since then AP bootstrap and GINX bootstrap have become the dominant bootstrap in the 3rd generation of FHE scheme [4-10]. The Compared with the 2nd generation, the 3rd generation FHE scheme has improved the performance of the bootstrap procedure and no longer needs to control the noise growth by using the dimension-mode reduction technique, which makes the computation more efficient.

The limitations are: 1) the security is reduced. 2) the realisation of full homomorphism is still not free from the bootstrap procedure, which requires a larger computational cost.

3.5. Fourth-Generation FHE Scheme Wwith Support for Floating-Point Arithmetic

The CKKS system, or fourth generation FHE scheme, was proposed by Cheon et al. in 2017. The approach for creating a translational homomorphic encryption scheme for roughly arithmetic numbers is proposed in this scheme, thus supporting floating-point arithmetic, and the workflow of this scheme is shown in Figure 2.

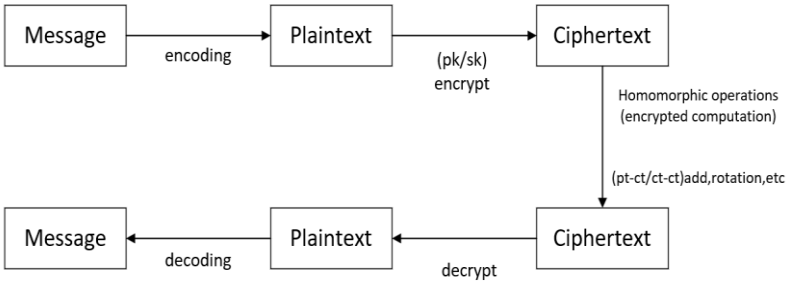

Figure 2. Flow of CKKS scheme implementation (Photo credit: Original).

However, the scheme was initially a SHE scheme, and a year later, Cheon et al. extended it to a FHE scheme by proposing a new approach for refreshing ciphertexts. The scheme works as follows:

Key generation: enter the security parameter λ and choose M along with the integers h and t, and the real numbers σ so as to achieve a complexity of 2^{λ}. Compute the secret key $_{\mathrm{sk}} = (1, s)$, where $s \in \chi$, randomly generate $a \in R_{q_L}$ and compute $-as + emodq_L$, and then compute $b' = -as + e' + ts'modt \cdot q_L$. Finally, output the key $_{\mathrm{sk}} = (1, s)$, public key $\mathrm{pk} = (b, a)$, and evaluation key $_{\mathrm{evk}} = (b', a')$.

Encryption: input message $m \in R$ and public key $\mathrm{pk} \in R_{q_L}^2$, randomly select $v \in ZO(1/2)$ 和 $e_0, e_1 \in DG(\sigma^2)$, and output c= $(\beta, \alpha) = v\mathrm{pk} + (m + e_0, e_1)modq_{\ell}$.

Decryption: input key $_{\mathrm{sk}} = (1, s)$ and $\mathbf{c} \in R_{q^{\ell}}^2$ and compute m= $\langle \mathbf{c}, \mathrm{sk} \rangle modq_{\ell} = \beta + \alpha smodq_{\ell}$.

Later on, Cheon et al. included a CRT-based ciphertext packing approach in their implementation of the CKKS scheme [11].

In 2021, Micciancio et al. launched a key recovery attack against CKKS, pointing out that the CKKS scheme could not fully capture passive attacks, and that an attacker could leak the key information by performing noise analysis on the decrypted message, thus requiring and proposing a security model that is stronger than the traditional IND-CPA security [12]. In the same year, Kim et al. proposed a bootstrap algorithm with better performance and more security against the above passive attacks by introducing a small rescaling error [13].

In 2022, Jutla and Manohar proposed a CKKS-based bootstrap of arbitrary accuracy by approximating the modulus function by a new low error sine series approximation to achieve a high accuracy bootstrap procedure and suggested that approximating this sinusoidal function and hence the modulus function by a Taylor series approximation can be used to achieve CKKS-based bootstrap of arbitrary accuracy [14].

In recent years, the research on FHE also generally focuses on optimising the bootstrap procedure of the CKKS scheme and the ciphertext conversion framework, so as to improve the computational efficiency and security.

The 4th generation FHE scheme is actually very similar to the 2nd generation FHE scheme, both of which have efficient packing techniques, while the primary distinction is that the 4th generation FHE scheme is an approximation scheme, which uses approximate computation and is much faster.

3.6. Comparative Analysis of Mainstream FHE Schemes

This section analyses the advantages and disadvantages of mainstream FHE schemes in terms of bootstrap performance and whether they support homomorphic operations. The results are shown in Table 1.

Table 1. The pros and cons of mainstream FHE schemes.

SCHEMES	Integer-based FHE	1st Generation	2nd Generation	3rd Generation	4th Generation
PROS	good security	support for batch processing	good security/ efficient bootstrapping	efficient bootstrapping/ simple calculation	efficient packing/ fast polynomial approx
CONS	computationally complex	bad security/ inefficient bootstrapping	large storage costs	bad security	inefficient bootstrapping

4. Applications of FHE

4.1. FHE in Cloud Computing

Encrypted database query: When storing encrypted databases in the cloud, users can perform encrypted queries using FHE technology. The encrypted query is processed by the cloud server, which returns the encrypted result without requiring the data to be decrypted. In this manner, the user's data and query requests remain encrypted throughout the process to ensure data privacy, and the whole process is shown in Figure. 3.

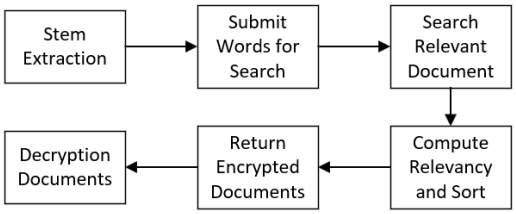

Figure 3. Data query process based on FHE (Photo credit: Original).

Encrypted data processing and analysis: in the financial field, some data and encrypted functions need to be kept confidential. For example, information such as company data, stock prices, performance, inventory lists, etc. are related to investment decisions and are key information for developing trading strategies, which can be encrypted and analysed by financial institutions in the cloud to ensure data security. In the medical field, patients' medical records and real-time monitoring data can be encrypted and transmitted to the cloud. Using FHE, cloud servers can compute these encrypted data to generate medical reports, diagnostic recommendations, etc., without accessing the patient's original data, ensuring that the patient's privacy is not compromised. Moreover, in some joint medical research, hospitals can encrypt patient data and conduct joint analyses in the cloud without exposing patient privacy.

4.2. FHE in Machine Learning

Federated Learning: A distributed machine learning strategy which allows multiple parties with data to train one or more models locally with no leakage of data from any one party to other participants is called federated learning, and then the model parameters are aggregated for global updating, whereas in combination with FHE, the privacy protection in Federated Learning can be further enhanced to boost the task performance of the participants' local models. In federated learning, joint training of models by multiple parties generally requires exchanging intermediate results, which may risk privacy leakage if explicit results are sent directly. In this scenario, FHE can play a very important role. Multiparties directly encrypt the intermediate results with homomorphic encryption algorithms, then send them to a third party for aggregation, and then return the aggregated results to all participants, which not only ensures that the intermediate results are not leaked, but also completes the training task. The federated learning process is shown in Figure. 4.

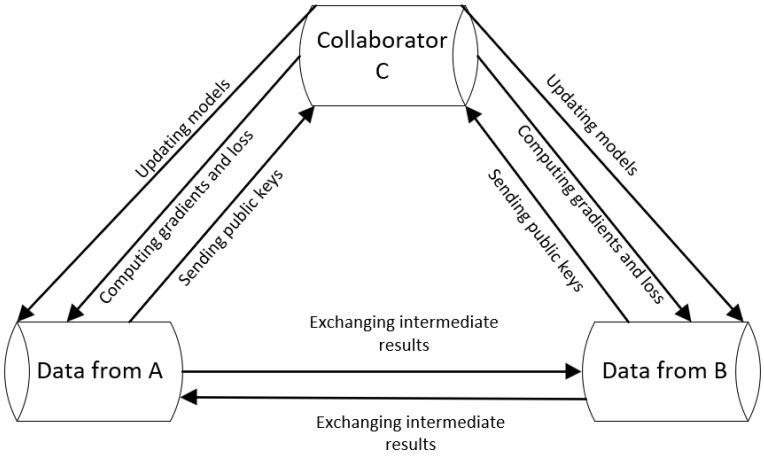

Figure 4. Example of a federal learning process (Photo credit: Original).

Privacy protection: in machine learning, data is the key to training models. However, if plaintext data is used directly for training, the user's privacy may be compromised. With FHE, data can be encrypted before training, thus protecting the user's privacy. At the same time, since FHE supports the computation of ciphertext and obtains the same results as unencrypted data, it can ensure that the training effect and accuracy of the model are not affected.

4.3. FHE in Electronic Voting

FHE (FHE) allows encrypted data to be computed without decryption, which means that ballots can be processed in an encrypted state, thus ensuring that the content of the voter's ballot is secure and private throughout the process. In addition, the ballot paper is encrypted during transmission and storage with the help of FHE, preventing anyone from tampering with the ballot paper without authorisation and ensuring the fairness of the ballot paper. And the FHE feature can provide a completely anonymous voting environment. It is significant to remember that FHE can also be combined with other cryptographic techniques to effectively prevent voter swiping.

5. Conclusion

The substantial applications of Fully Homomorphic Encryption (FHE) in domains such as cloud computing, machine learning, and e-voting have positioned it as a cornerstone in the field of cryptography, attracting increasing attention from cryptographers. Consequently, research aimed at optimizing and enhancing FHE has progressed rapidly in recent years, achieving significant breakthroughs in computational efficiency, accuracy, security, and expanding application domains. As the breadth and depth of these applications grow, the demands on FHE-related technologies continue to escalate.

In light of the advancements in FHE schemes discussed previously, it is recommended that future research should concentrate on three key areas:

1. Further optimization of the speed and accuracy of the bootstrapping procedure, or alternatively, exploring avenues to eliminate the need for bootstrapping entirely.

2. Rigorous validation of the security assumptions underlying existing FHE schemes to reinforce their reliability and robustness.

3. Development of innovative FHE schemes capable of integrating the functionalities of the second, third, and fourth generation FHE architectures.

These focal points aim to address the evolving requirements of secure computing environments, ensuring that FHE remains at the forefront of cryptographic research and practical application.

References

[1] Z. Brakerski, C. Gentry, V. Vaikuntanathan, (Leveled) FHE without bootstrap, ACM Trans. Comput. Theory 6(3) (2014) 1-36.

[2] Z. Brakerski, FHE without modulus switching from classical GapSVP, in: Annu. Cryptol. Conf., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 868-886.

[3] J. Fan, F. Vercauteren, Somewhat practical FHE, Cryptol. ePrint Arch. (2012).

[4] S. Garg, D. Gupta, Efficient round optimal blind signatures, in: Adv. Cryptol.–EUROCRYPT 2014: 33rd Annu. Int. Conf. on the Theory and Appl. of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014, Proc. 33, Springer Berlin Heidelberg, 2014, pp. 477-495.

[5] I.Chillotti, N. Gama, M. Georgieva, M. Izabachene, Faster FHE: Bootstrap in less than 0.1 seconds, in: Adv. Cryptol.–ASIACRYPT 2016: 22nd Int. Conf. on the Theory and Appl. of Cryptology and Inf. Secur., Hanoi, Vietnam, Dec. 4-8, 2016, Proc., Part I 22, Springer Berlin Heidelberg, 2016, pp. 3-33.

[6] I.Chillotti, N. Gama, M. Georgieva, M. Izabachene, Faster packed homomorphic operations and efficient circuit bootstrap for TFHE, in: Int. Conf. on the Theory and Appl. of Cryptology and Inf. Secur., Cham: Springer Int. Publ., 2017, pp. 377-408.

[7] Chillotti, N. Gama, M. Georgieva, M. Izabachene, TFHE: fast FHE over the torus, J. Cryptol. 33(1) (2020) 34-91.

[8] J. Alperin-Sheriff, C. Peikert, Faster bootstrap with polynomial error, in: Adv. Cryptol.–CRYPTO 2014: 34th Annu. Cryptol. Conf., Santa Barbara, CA, USA, Aug. 17-21, 2014, Proc., Part I 34, Springer Berlin Heidelberg, 2014, pp. 297-314.

[9] N. Gama, et al., Structural lattice reduction: generalized worst-case to average-case reductions and homomorphic cryptosystems, in: Adv. Cryptol.–EUROCRYPT 2016: 35th Annu. Int. Conf. on the Theory and Appl. of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proc., Part II 35, Springer Berlin Heidelberg, 2016.

[10] J.H. Cheon, K. Han, A. Kim, M. Kim, Y. Song, A full RNS variant of approximate homomorphic encryption, in: Sel. Areas in Cryptography–SAC 2018: 25th Int. Conf., Calgary, AB, Canada, Aug. 15–17, 2018, Rev. Sel. Pap., 25, Springer Int. Publ., 2019, pp. 347-368.

[11] B. Li, D. Micciancio, On the security of homomorphic encryption on approximate numbers, in: Annu. Int. Conf. on the Theory and Appl. of Cryptographic Techniques, Cham: Springer Int. Publ., 2021, pp. 648-677.

[12] A.Kim, M. Deryabin, J. Eom, R. Choi, Y. Lee, W. Ghang, D. Yoo, General bootstrap approach for RLWE-based homomorphic encryption, IEEE Trans. Comput. (2023).

[13] C.S. Jutla, N. Manohar, Sine series approximation of the mod function for bootstrap of approximate HE, in: Annu. Int. Conf. on the Theory and Appl. of Cryptographic Techniques, Cham: Springer Int. Publ., 2022, pp. 491-520.

[14] M. Naehrig, K. Lauter, V. Vaikuntanathan, Can homomorphic encryption be practical? in: Proc. of the 3rd ACM workshop on Cloud computing security workshop, 2011, pp. 113-124.