Fully Homomorphic Encryption for Secure Cloud Computation

Sreekutty Sabarivasan[1]; Dr. Ashish L.[2]

[1]MCA Scholar, [2]Assistant Professor

[1;2]Department of MCA, Nehru College of Engineering and Research Centre, Thrissur, India

Publication Date: 2026/03/07

Abstract: Fully Homomorphic Encryption (FHE) is an advanced cryptographic technique that enables computation on encrypted data without requiring decryption. This capability eliminates the need to expose sensitive data during processing, making FHE particularly suitable for cloud computing, healthcare analytics, financial systems, and privacy-preserving artificial intelligence. Despite its strong theoretical foundation, FHE faces practical challenges including high computational complexity, large ciphertext expansion, and bootstrapping overhead. This paper presents a comprehensive study of FHE, including its theoretical background, working mechanism, security properties, real-world applications, limitations, and emerging research trends.

Keywords: *Fully Homomorphic Encryption, Cloud Security, Bootstrapping, Lattice Cryptography, Privacy-Preserving Computation.*

How to Cite: Sreekutty Sabarivasan; Dr. Ashish L. (2026) Fully Homomorphic Encryption for Secure Cloud Computation. *International Journal of Innovative Science and Research Technology*, 11(2), 2794-2796. https://doi.org/10.38124/ijisrt/26feb1493

I. INTRODUCTION

Cloud computing has transformed modern data processing, but outsourcing sensitive data to third-party servers introduces serious privacy concerns. Traditional encryption protects data at rest and in transit but requires decryption before computation. Fully Homomorphic Encryption (FHE) overcomes this limitation by enabling computation directly on encrypted data, making it a promising solution for secure cloud computing and confidential analytics.

II. TYPES OF HOMOMORPHIC ENCRYPTION

Fully Homomorphic Encryption (FHE) represents the most powerful category, enabling arbitrary and unlimited computations on encrypted data. This capability is achieved through bootstrapping techniques that periodically refresh ciphertexts to control noise growth.

Somewhat Homomorphic Encryption (SHE) extends this capability by supporting both addition and multiplication, but only up to a bounded depth due to noise accumulation within ciphertexts.

Homomorphic encryption schemes are broadly classified based on the type and number of algebraic operations they support on encrypted data. Partially Homomorphic Encryption (PHE) permits unlimited execution of a single mathematical operation, such as addition or multiplication, making it useful for limited cryptographic applications.

Homomorphic encryption is categorized based on its mathematical capacity and the number of operations it can perform on encrypted data.

- Partially Homomorphic Encryption (PHE): Supports only one type of operation, such as addition or multiplication, for an unlimited number of times.
- Somewhat Homomorphic Encryption (SHE): Supports both addition and multiplication but is limited to a pre-defined number of operations before the "noise" prevents successful decryption.
- Fully Homomorphic Encryption (FHE): Enables unlimited arithmetic operations of any complexity using bootstrapping to manage noise growth.

III. WORKING MECHANISM

Finally, the authorized data owner decrypts the resulting ciphertext using the private key to obtain the correct computation result.

To manage noise growth, bootstrapping techniques are applied to refresh ciphertexts without revealing the underlying plaintext, thereby enabling unlimited computation.

Homomorphic evaluation allows cloud servers or third-party entities to perform computations directly on ciphertexts. Each operation increases the internal noise level.

During encryption, plaintext data is transformed into ciphertext using complex algebraic structures that deliberately introduce controlled noise to guarantee semantic security.

The working mechanism of Fully Homomorphic Encryption is designed to ensure complete data confidentiality throughout the computation lifecycle. The process begins with key generation, which produces public keys for encryption, private keys for decryption, and evaluation keys used during encrypted computation.

The FHE workflow is a multi-stage process designed to maintain data integrity while obscured.

- Key Generation: Involves creating the public key for encryption, the secret key for decryption, and evaluation keys for performing operations.
- Encryption: The plaintext is converted into ciphertext using a mathematical structure that includes a small "noise" component to ensure security.

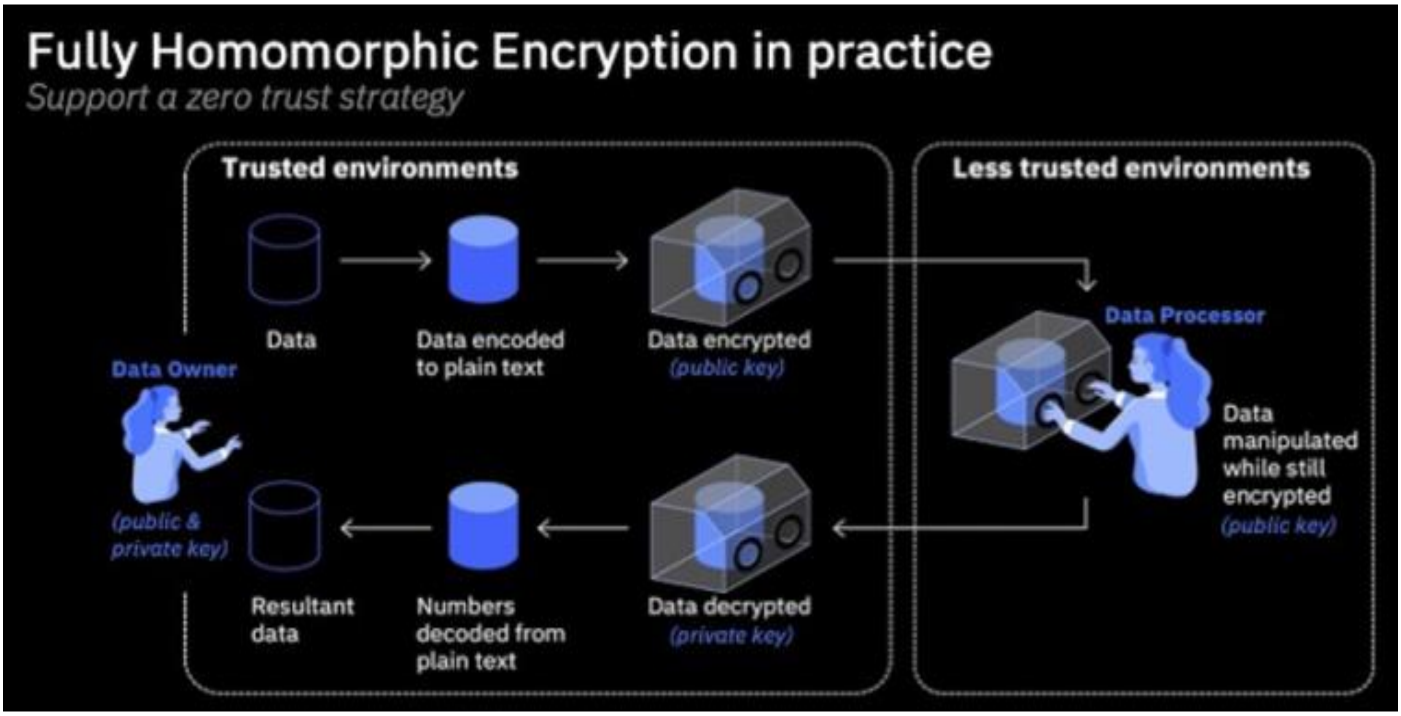
Fully Homomorphic Encryption in practice
Support a zero trust strategy
Trusted environments
Less trusted environments
Data
Data encoded to plain text
Data encrypted (public key)
Data Owner
(public & private key)
Data Processor
Data manipulated while still encrypted (public key)
Resultant data
Numbers decoded from plain text
Data decrypted (private key)

Fig 1 Fully Homomorphic Encryption in Practice

- Homomorphic Evaluation: A third-party applies functions to the ciphertext, which increases the internal noise with each operation.
- Noise Management (Bootstrapping): Periodically "refreshes" the ciphertext to reduce noise without revealing the original plaintext.
- Final Decryption: The end-user uses their private key to reveal the results of the computation

IV. APPLICATIONS

FHE can be used in many areas where keeping data private is important. In secure cloud computing, it lets you outsource processing tasks without giving the provider access to raw data. This gets rid of the need to trust the provider, which is a big reason why cloud computing isn't more widely used for sensitive workloads. FHE is very helpful for healthcare analytics because it lets you look at sensitive patient records or genomic data while still following HIPAA privacy rules and data protection laws. Researchers can work together to analyze distributed medical datasets without revealing any private patient information. This speeds up medical discoveries while keeping patient information private.

Financial services can implement secure fraud detection and credit scoring on encrypted transaction data, enabling real-time risk assessment without exposing customer financial details to unauthorized parties. Additionally, FHE facilitates privacy-preserving machine learning, enabling the training of AI models on sensitive datasets without exposing raw inputs. This capability is particularly valuable for industries such as finance, healthcare, and defense, where data sensitivity often prevents organizations from leveraging cloud-based AI services. The technology also extends to IoT protection, safeguarding data generated by Internet of Things devices during both transit and processing, thereby addressing security vulnerabilities in smart home, industrial IoT, and connected vehicle applications.

V. ADVANTAGES

FHE offers several compelling advantages for secure computation. It provides end-to-end confidentiality, ensuring strong protection throughout the entire data lifecycle from encryption through processing to final decryption. The technology enables trustless security by eliminating the need to trust cloud providers with sensitive plaintext data, fundamentally changing the security model for outsourced computation. Unlike simpler encryption schemes, FHE supports arbitrary computation, allowing any mathematical function or algorithm to be executed on encrypted bits, making it universally applicable across domains. Furthermore, many FHE implementations offer post-quantum security due to their lattice-based foundations, providing protection against future quantum-level threats that could compromise traditional public-key cryptosystems. This combination of comprehensive confidentiality and future-proof security makes FHE uniquely positioned to address emerging privacy challenges.

VI. LIMITATIONS

Despite its theoretical promise, FHE faces significant practical limitations. Computational overhead remains a major challenge, as processing encrypted data is substantially slower than processing plaintext. Ciphertext expansion poses another obstacle, with encrypted data often being thousands of times larger than the original plaintext. Additionally, high memory requirements and computationally expensive bootstrapping operations make real-time applications particularly challenging to implement with current FHE technology.

VII. FUTURE SCOPE

Future research is focused on transitioning FHE into a practical standard for real-world deployment. Hardware acceleration through GPUs, FPGAs, and specialized processors such as CraterLake shows promise in speeding up FHE computations significantly, with some implementations achieving orders of magnitude performance improvements over software-only approaches. Researchers are also developing lightweight FHE variants designed for edge devices and resource-constrained sensors, enabling privacy-preserving computation at the network edge. Furthermore, advancements in AI integration aim to create energy-efficient encrypted computation methods suitable for large-scale artificial intelligence applications, including confidential inference and privacy-preserving model training. Standardization efforts led by organizations such as the Homomorphic Encryption Standardization Consortium are establishing common benchmarks, security parameters, and interoperability protocols that will facilitate broader adoption.

VIII. CONCLUSION

Fully Homomorphic Encryption represents a transformative technology for privacy-preserving computation in the cloud and beyond. By enabling direct computation on encrypted data without requiring decryption, FHE addresses fundamental privacy challenges that have historically limited the adoption of cloud computing for sensitive workloads. The technology's ability to provide end-to-end confidentiality while supporting arbitrary computations positions it as a cornerstone of future privacy-centric digital infrastructure.

The journey from theoretical concept to practical implementation has been marked by significant milestones, beginning with Gentry's groundbreaking bootstrap technique in 2009 and continuing through subsequent innovations in scheme efficiency, security, and usability. Modern FHE implementations benefit from decades of cryptographic research, lattice-based hardness assumptions, and ongoing optimization efforts that have progressively reduced the performance gap between encrypted and plaintext computation.

REFERENCES

[1]. C. Gentry, Fully homomorphic encryption using ideal lattices, *STOC*, Vol. 41, 2009.
[2]. Z. Brakerski et al., Leveled fully homomorphic encryption without bootstrapping, *ITCS*, Vol. 15, 2014.
[3]. J. H. Cheon et al., Homomorphic encryption for approximate numbers, *ASIACRYPT*, Vol. 10624, 2017.
[4]. C. Marcolla et al., Survey on fully homomorphic encryption, *Journal of Cryptology*, Vol. 35, 2022.
[5]. C. Gentry and S. Halevi, Implementing FHE scheme, *EUROCRYPT*, Vol. 6632, 2011.
[6]. J. Fan and F. Vercauteren, Somewhat practical FHE, *IACR ePrint*, Vol. 2012/144, 2012.
[7]. N. Samardzic et al., F1 accelerator for FHE, *MICRO*, Vol. 54, 2021.
[8]. N. Samardzic et al., CraterLake accelerator, *ISCA*, Vol. 49, 2022.
[9]. P. Fauzi et al., IND-CCA1 security of FHE, *PKC*, Vol. 13178, 2022.
[10]. H. Shen et al., Bootstrapping survey, *IEEE Surveys & Tutorials*, Vol. 27, 2025.
[11]. A. Akavia and M. Vald, Privacy of homomorphic protocols, *TCC*, Vol. 20042, 2021.
[12]. I. Chillotti et al., Attacking FHE applications, *FHE Standards Workshop*, Vol. 3, 2016.
[13]. S. Goldwasser et al., Reusable garbled circuits, *FOCS*, Vol. 54, 2013.
[14]. M. Albrecht et al., Homomorphic encryption security standard, *HomomorphicEncryption.org*, Vol. 1, 2018.
[15]. S. Angel et al., PIR with compressed queries, *SIGMOD*, Vol. 47, 2018..