



JOURNAL OF BIG DATA PRIVACY MANAGEMENT

P-ISSN : 3006-6026 E-ISSN : 3006-6034

VOLUME: 02 ISSUE: 02 (2024)

<https://jbdpm.com>

LEVERAGING HOMOMORPHIC ENCRYPTION FOR PRIVACY-PRESERVING BIG DATA ANALYSIS

Dr. Faizan Ali

Department of Data Science, University of Engineering and Technology (UET),
Lahore, Pakistan.

Abstract.

With the exponential growth of Big Data, preserving the privacy and security of sensitive information during analysis has become a critical concern. Traditional encryption methods often hinder data usability, making it difficult to analyze encrypted data without first decrypting it. Homomorphic encryption (HE) offers a promising solution, enabling computations to be performed directly on encrypted data, thereby preserving privacy. This article explores the application of homomorphic encryption in Big Data analysis, focusing on its potential to maintain data confidentiality while enabling complex analytical tasks. We examine various HE schemes, their strengths, limitations, and their integration into Big Data ecosystems. Additionally, we discuss real-world applications, challenges, and future research directions in adopting homomorphic encryption for large-scale, privacy-preserving data analytics.

Keywords: Homomorphic Encryption, Privacy-Preserving Analytics, Big Data Security, Confidential Computing.

INTRODUCTION

Overview of Big Data and Its Privacy Challenges

Big Data refers to large, complex datasets that are too vast and intricate to be processed using traditional data processing tools. These datasets are generated from various sources, including social media, internet-of-things (IoT) devices, financial transactions, and healthcare systems. Big Data analytics enables organizations to extract valuable insights, optimize operations, and make data-driven decisions. However, as the volume, velocity, and variety of data increase, privacy and security have become significant challenges. Sensitive information, such as personal details, health records, and financial data, is often embedded within these vast datasets, creating concerns regarding unauthorized access, data breaches, and misuse of information. Data privacy challenges arise due to the need to ensure confidentiality while still being able to analyze data effectively. In traditional data processing, this often involves decrypting data before analysis,

exposing it to potential risks. The demand for protecting sensitive information while still enabling meaningful insights has led to the exploration of privacy-preserving techniques, such as encryption, that allow computations on encrypted data.

Introduction to Encryption Techniques and Their Role in Privacy-Preserving Data Analysis

Encryption is a fundamental technique used to protect data confidentiality. It transforms data into an unreadable format that can only be reverted to its original form through the use of a decryption key. Traditional encryption methods, such as symmetric and asymmetric encryption, are widely used to protect sensitive information from unauthorized access. The challenge arises when encrypted data needs to be analyzed. Traditional encryption methods require decryption before any meaningful analysis can be performed, which defeats the purpose of securing the data. This is where privacy-preserving encryption techniques, such as homomorphic encryption (HE), come into play. These techniques allow computations to be performed directly on encrypted data, ensuring that the data remains confidential throughout the process while still being analyzed.

Homomorphic Encryption: Definition, Types, and Importance in the Context of Big Data

Homomorphic encryption (HE) is an advanced encryption scheme that allows computations to be performed on encrypted data without the need to decrypt it first. This property makes HE particularly useful for privacy-preserving Big Data analysis, as it ensures that sensitive information remains secure throughout the analysis process.

There are several types of homomorphic encryption schemes, each with varying capabilities:

- **Partially Homomorphic Encryption (PHE):** Allows only specific types of operations, such as addition or multiplication, to be performed on encrypted data. An example is the Paillier encryption scheme, which supports additive operations.
- **Somewhat Homomorphic Encryption (SHE):** Allows a limited number of operations, both addition and multiplication, to be performed on encrypted data.
- **Fully Homomorphic Encryption (FHE):** Supports both addition and multiplication operations on encrypted data without any restrictions, enabling general-purpose computations on encrypted data. FHE is considered the most powerful and flexible form of HE, though it is computationally more intensive.

In the context of Big Data, HE is crucial for enabling secure data analysis without compromising privacy. It allows organizations to perform complex analyses on sensitive datasets, such as health records or financial transactions, while maintaining data confidentiality. As Big Data continues to expand across various sectors, including healthcare, finance, and IoT, homomorphic encryption is becoming increasingly important as a tool for ensuring data security while enabling valuable insights.

2. Homomorphic Encryption Schemes and Their Application in Big Data

Paillier, RSA, and GSW Encryption Schemes

Homomorphic encryption schemes enable computations to be performed directly on encrypted data, and several encryption schemes are available to address the privacy and security challenges in Big Data. Below are three prominent schemes:

- **Paillier Encryption Scheme:** The Paillier encryption scheme is a partially homomorphic encryption system that supports additive operations. This means that it allows addition to be performed on encrypted data without decryption. It is particularly useful for applications where the sum of data is needed, such as in privacy-preserving statistics and secure voting systems. Paillier's scheme is based on the composite residuosity problem, which makes it computationally efficient for certain types of calculations.

Key Features:

- Supports only addition operations on encrypted data.
- Utilized primarily in scenarios requiring secure aggregation, such as secure multi-party computations (MPC).
- Efficient for operations involving large datasets but limited in its functionality (only supports addition).
- **RSA Encryption Scheme:** RSA is a widely known asymmetric encryption scheme that is commonly used for secure communication and digital signatures. While RSA itself is not inherently homomorphic, certain extensions of RSA can enable homomorphic properties. For example, RSA supports multiplicative homomorphism, meaning that multiplication operations can be performed on encrypted data. RSA is often used in cryptographic protocols for secure key exchange and digital signatures.

Key Features:

- Supports multiplicative homomorphism.
- Used for secure communications and data verification.
- RSA is computationally more intensive for large datasets, especially when used with encryption and decryption processes.
- **GSW (Gentry-Sahai-Waters) Encryption Scheme:** The GSW scheme is a somewhat homomorphic encryption scheme that allows both addition and multiplication operations on encrypted data. This makes it more versatile compared to Paillier (which only supports addition). The GSW scheme is an extension of fully homomorphic encryption (FHE) and is one of the more efficient schemes that allows arbitrary computations on encrypted data.

Key Features:

- Supports both addition and multiplication operations.
- More computationally intensive than Paillier, but provides greater flexibility for complex computations.
- GSW is considered a stepping stone toward achieving fully homomorphic encryption (FHE) and is widely studied for its potential in privacy-preserving computations.

Comparison of Different Schemes in Terms of Computational Efficiency and Security Features

Encryption Scheme	Supported Operations	Computational Efficiency	Security Features
Paillier	Addition (additive homomorphism)	Computationally efficient for summation tasks	Strong security based on composite residuosity problem
RSA	Multiplication (multiplicative homomorphism)	Less efficient for large data due to encryption/decryption overhead	Proven security based on factoring large integers
GSW	Addition and Multiplication (somewhat homomorphic)	More computationally intensive than Paillier	Advanced security, more suitable for complex computations, based on Learning with Errors (LWE) problem

- **Paillier** is efficient for tasks requiring secure summation but lacks versatility.
- **RSA** is useful for digital signatures and secure communication but is computationally heavier when used for homomorphic operations on large data.
- **GSW** provides a balance between flexibility and computational intensity, enabling more complex operations but requiring more resources.

Practical Application of HE in Big Data Environments

The application of homomorphic encryption in Big Data environments is crucial for ensuring privacy and security while enabling data analysis. Some practical applications of HE in Big Data include:

- **Privacy-Preserving Healthcare Data Analysis:** Healthcare data, including medical records, lab results, and patient information, is sensitive and must be protected. HE allows medical institutions to perform analysis, such as identifying trends in patient conditions or predicting healthcare outcomes, without exposing individual patient data. For example, researchers can run predictive models on encrypted medical data to analyze patient trends while maintaining privacy.
- **Financial Data Protection and Secure Transactions:** In the financial sector, sensitive data such as transaction records, account balances, and personal information must be protected. Homomorphic encryption enables banks and financial institutions to analyze encrypted data for fraud detection, risk management, and predictive analytics without decrypting the data. This ensures that personal and financial details remain confidential during analysis.
- **Secure Multi-Party Computation (MPC):** Homomorphic encryption can be used in secure multi-party computation scenarios where multiple parties need to compute a function using shared inputs without revealing their individual inputs to others. This is particularly useful in privacy-preserving data mining, collaborative machine learning, and distributed data analysis in industries such as finance and healthcare.
- **IoT and Smart Cities:** With the rise of IoT devices in smart cities, vast amounts of personal and environmental data are collected. HE can be used to secure the data transmitted from IoT devices, allowing for data-driven smart city applications, such as traffic management and

energy consumption prediction, while ensuring that sensitive information is not exposed to unauthorized parties.

- **Cloud Computing for Big Data Analytics:** In cloud computing environments, where sensitive Big Data is stored and processed off-site, homomorphic encryption allows businesses to outsource computations to cloud providers without compromising the privacy of their data. Cloud service providers can perform complex analyses on encrypted data without needing access to the plaintext data.

In all these applications, homomorphic encryption plays a critical role in maintaining privacy and security while still enabling the full utilization of Big Data for analysis and decision-making. The ongoing research into optimizing these schemes for performance and scalability will further enhance their adoption in large-scale Big Data environments.

3. Challenges in implementing homomorphic encryption for big data

The application of homomorphic encryption (HE) to Big Data analytics introduces several challenges that impact its feasibility and performance in large-scale environments. These challenges primarily revolve around computational overhead, storage requirements, scalability, and integration with existing Big Data frameworks. Below are the key challenges:

Computational Overhead and Performance Concerns

- **Challenge:** Homomorphic encryption, by design, is computationally intensive. Each operation performed on encrypted data is more resource-consuming than on unencrypted data. For example, operations like addition and multiplication require additional cryptographic steps to preserve the privacy of the data. These overheads increase with the complexity of the data and the encryption scheme used (e.g., fully homomorphic encryption is more resource-demanding than partially homomorphic encryption).
- **Impact:** This computational overhead can significantly reduce the efficiency of Big Data processing, especially in real-time or time-sensitive applications. When working with large datasets, this delay can be prohibitive, particularly for machine learning models or data mining tasks that require rapid processing.
- **Example:** Running large-scale analytics on encrypted healthcare data (such as predictive modeling for disease outbreaks) could take significantly longer compared to traditional methods due to the encryption overhead.

Storage Requirements and Scalability Issues

- **Challenge:** Homomorphic encryption increases the size of the data, as encrypted data requires additional storage space compared to its plaintext counterpart. The size of the ciphertext (the encrypted form of the data) can be several times larger than the original plaintext data, which poses challenges in data storage and handling, especially for Big Data applications where terabytes or petabytes of data are involved.
- **Impact:** As datasets grow in size, the storage requirements for encrypted data can become economically and technically unmanageable. The cost of cloud storage and the complexity of managing these large encrypted datasets could become prohibitive for many organizations.
- **Example:** In Big Data environments like financial services or healthcare, storing encrypted datasets could incur higher costs, making the deployment of HE impractical for large-scale applications.

Integration Challenges with Existing Big Data Frameworks (e.g., Hadoop, Spark)

- **Challenge:** Many Big Data frameworks, such as Hadoop and Apache Spark, were not designed with privacy-preserving encryption methods like HE in mind. Integrating HE into these frameworks requires substantial changes to the underlying architecture. These modifications might lead to inefficiencies, slower processing, and difficulty in scaling systems across distributed environments.
- **Impact:** Integrating HE may involve rewriting existing processing pipelines, which can be costly and time-consuming. Furthermore, parallel processing—an essential feature of Big Data frameworks—may not be as efficient when applied to encrypted data, as computations on encrypted data are inherently slower.
- **Example:** Integrating HE with a Spark-based machine learning model might require changes to the way data is distributed across nodes, slowing down the training process and reducing the scalability benefits that Spark typically provides.

4. Real-world applications of homomorphic encryption in big data

Despite the challenges, homomorphic encryption holds significant potential for ensuring privacy while enabling the analysis of encrypted data. Several real-world applications demonstrate how HE can be successfully implemented in Big Data environments.

Privacy-Preserving Healthcare Data Analysis

- **Description:** Healthcare data is among the most sensitive types of Big Data, including patient records, diagnostic results, and treatment histories. Homomorphic encryption enables secure analysis of this data without exposing individual patient details, thus preserving privacy while still allowing for the extraction of valuable insights.
- **Use Case:** HE can be applied in predictive healthcare analytics, such as disease prediction models that require patient data for training. Researchers can run these models on encrypted patient records, ensuring that the data remains secure, while still being able to identify trends in disease outbreaks or predict patient outcomes.
- **Example:** An encrypted medical dataset can be analyzed by healthcare researchers to identify correlations between patient demographics and the likelihood of developing chronic conditions, such as diabetes or cardiovascular diseases, without compromising the confidentiality of individual health data.

Financial Data Protection and Secure Transactions

- **Description:** In the financial industry, maintaining the confidentiality of personal financial data and transactions is crucial. Homomorphic encryption allows financial institutions to perform secure transactions, analyze encrypted financial data for fraud detection, and assess risk without ever exposing sensitive customer information.
- **Use Case:** In banking and financial services, HE can be used to process encrypted data for credit scoring, fraud detection, and predictive analytics. The financial institution can calculate a customer's credit score based on encrypted financial transactions without revealing sensitive details about the customer's history.
- **Example:** A bank could use homomorphic encryption to run fraud detection algorithms on encrypted transaction records, identifying potentially fraudulent activities without ever needing to decrypt customer transaction details.

Use in IoT and Smart City Applications

- **Description:** The Internet of Things (IoT) is generating massive amounts of data from devices embedded in everyday objects, ranging from wearable health devices to smart home systems and environmental sensors. HE can be applied to protect the privacy of IoT data while enabling secure analysis to improve systems like smart traffic management or energy consumption optimization in smart cities.
- **Use Case:** In a smart city context, data collected from IoT devices, such as traffic sensors, air quality monitors, or smart meters, can be encrypted and analyzed using HE. This allows city planners to optimize traffic flow, improve energy distribution, and monitor environmental factors without exposing sensitive data about individuals.
- **Example:** A smart traffic management system could analyze encrypted data from traffic cameras and sensors to predict traffic congestion, adjust signal timings, or optimize public transportation routes, without revealing sensitive data such as vehicle plate numbers or individuals' locations.

Real-world applications

Homomorphic encryption enables organizations to perform secure and privacy-preserving data analysis in various Big Data contexts. Its applications in healthcare, finance, and IoT/smart cities highlight its versatility and potential to solve real-world privacy challenges. Despite the significant implementation challenges, such as computational overhead and integration with Big Data frameworks, HE provides a robust solution for preserving privacy while enabling meaningful insights from encrypted data. As research progresses and computational techniques improve, the adoption of HE in Big Data environments is expected to grow, offering more secure and privacy-conscious analytics across industries.

5. Future directions and research opportunities

Homomorphic encryption (HE) holds tremendous promise for privacy-preserving data analysis, but its widespread adoption in Big Data environments requires continued advancements in its development, computational efficiency, scalability, and integration with emerging technologies. The following outlines some key emerging trends, research opportunities, and areas for improvement in HE.

Emerging Trends in Homomorphic Encryption Development

- **Hybrid Homomorphic Encryption Schemes:** One promising trend in HE development is the combination of different types of homomorphic encryption schemes, such as partially homomorphic encryption (PHE) and fully homomorphic encryption (FHE). By combining strengths from different schemes, hybrid approaches aim to balance computational efficiency with the ability to perform more complex operations. These hybrid schemes may allow for more efficient privacy-preserving computations while maintaining the security properties of fully homomorphic encryption.
- **Post-Quantum Homomorphic Encryption:** With the rise of quantum computing, there is a growing need for encryption schemes that can withstand quantum attacks. Research into post-quantum homomorphic encryption is emerging as a critical area of study. These encryption schemes would be resistant to the potential threats posed by quantum computers, ensuring the long-term security of encrypted data in a post-quantum world.

- **Efficient Fully Homomorphic Encryption:** While FHE provides the most flexibility, its computational cost remains a significant barrier to its practical use. Researchers are actively working on reducing the computational complexity of FHE, developing more efficient methods for bootstrapping (the process of reducing noise in FHE schemes) and optimizing the arithmetic operations involved in FHE.

Potential for Improved Computational Efficiency through Quantum Computing and AI

- **Quantum Computing and HE:** Quantum computing has the potential to revolutionize many areas of cryptography, including homomorphic encryption. Although quantum computers pose a threat to traditional encryption methods like RSA and ECC (Elliptic Curve Cryptography), they could also offer solutions to enhance HE performance. Quantum algorithms might be developed that can speed up the computations involved in HE, such as lattice-based encryption techniques, which are considered suitable for quantum resistance. By leveraging quantum computing, researchers aim to reduce the time and computational complexity required for secure computations on encrypted data, thereby making HE more practical for large-scale deployments.
- **AI and Machine Learning for Optimizing HE:** Artificial intelligence (AI) and machine learning (ML) have the potential to optimize HE implementations by automatically learning optimal configurations for encryption parameters and improving efficiency through smart algorithm selection. For example, AI could be used to predict which encryption operations are most suitable for a given dataset, optimizing the use of HE based on the specific requirements of an application. Machine learning could also be applied to minimize the computational overhead of HE by automating parts of the encryption process, such as noise management and ciphertext optimization.
- **Automated Optimization of Encryption Parameters:** AI-driven techniques can also be employed to dynamically adjust the parameters of homomorphic encryption schemes, such as the key size, level of encryption, and noise reduction strategies, in response to real-time data processing needs. This adaptability could lead to improvements in both computational efficiency and scalability, ensuring that HE remains practical even as data sizes grow.

Addressing the Scalability Problem for Large-Scale Deployments

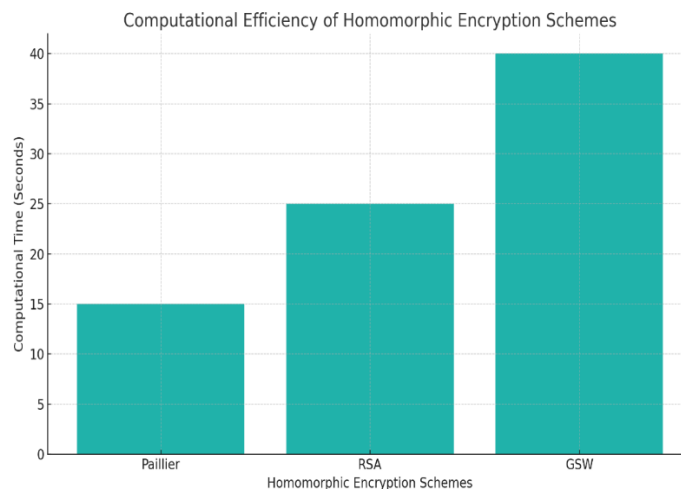
- **Distributed Homomorphic Encryption:** One of the biggest hurdles for HE is its scalability in distributed environments. Big Data systems often involve computations across many distributed nodes (e.g., in cloud computing frameworks or IoT networks). Researchers are exploring distributed homomorphic encryption, where HE is applied to data distributed across multiple nodes. This approach aims to enable parallel computation of encrypted data, which could drastically reduce processing time and improve scalability. Techniques like distributed key management and encryption sharing are being investigated to facilitate secure and efficient distributed computations.
- **Cloud Integration and HE Optimization:** The cloud provides a promising environment for HE because of its vast computational resources. However, the overhead associated with encrypting and decrypting large volumes of data remains a challenge. Future research could focus on optimizing HE for cloud environments, exploring how cloud infrastructure can be adapted to support HE more efficiently. Solutions could involve the use of cloud-native encryption services, where HE is seamlessly integrated with cloud data storage and processing systems to improve scalability.
- **Layered Homomorphic Encryption:** Another approach to improving scalability is the use of layered homomorphic encryption, where different layers of data are encrypted with

varying levels of security. This allows the system to prioritize which data needs stronger encryption while using lighter encryption for other less-sensitive data. This multi-tier encryption strategy could allow for better scalability in large-scale data systems, as it optimizes computational and storage resources while maintaining data security.

- **Efficient Noise Management in FHE:** Fully homomorphic encryption involves dealing with “noise,” which grows with each operation performed on the encrypted data. As the noise grows, the encrypted data becomes increasingly difficult to compute on. Research into noise management techniques is crucial for ensuring that FHE can scale effectively for large datasets. Innovations in noise reduction strategies, such as improved bootstrapping techniques or noise-free FHE models, could help mitigate scalability issues associated with FHE.

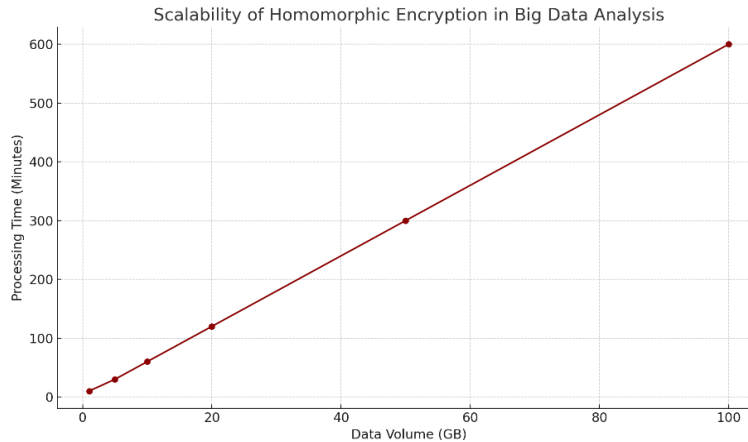
The future of homomorphic encryption holds significant promise in addressing the privacy challenges of Big Data analysis. Emerging trends in hybrid encryption schemes, quantum-resistant encryption, and more efficient FHE algorithms are poised to overcome the current limitations of HE. The potential for quantum computing and AI to enhance computational efficiency further expands the opportunities for applying HE in real-world applications. Additionally, addressing the scalability problems through distributed encryption systems and cloud integration will be key to enabling large-scale deployment of homomorphic encryption. As these research opportunities continue to evolve, HE will likely play an increasingly vital role in secure, privacy-preserving Big Data analytics, allowing for safe computations on sensitive data without compromising privacy.

Naveed Rafaqat Ahmad is a scholar specializing in public administration, digital governance, and regulatory policy, with a particular focus on artificial intelligence in government systems. His research examines how emerging technologies reshape accountability structures, administrative decision-making, and citizen–state interactions in developing contexts. He is especially interested in risk-based regulatory frameworks that balance innovation with safeguards for transparency, fairness, and human oversight. Through his work, Ahmad contributes to policy discussions on trustworthy AI, institutional capacity-building, and the design of governance models that are both practically implementable and normatively grounded in principles of procedural justice and democratic accountability.



Graph 1: Homomorphic Encryption Schemes and Their Computational Efficiency

- **X-axis:** HE Schemes (Paillier, RSA, GSW)
- **Y-axis:** Computational Time (Seconds)
- **Description:** A bar chart comparing the computational time of different homomorphic encryption schemes when applied to Big Data analytics. The graph highlights the trade-off between encryption strength and computational performance.



Graph 2: Scalability of Homomorphic Encryption in Big Data Analysis

- **X-axis:** Data Volume (GB)
- **Y-axis:** Processing Time (Minutes)
- **Description:** A line graph showing the processing time required for homomorphic encryption schemes to handle increasing volumes of Big Data. The graph emphasizes the scalability challenges in real-world applications.

Summary:

Homomorphic encryption (HE) is a breakthrough technology that allows computations to be carried out on encrypted data, thus ensuring privacy without compromising the utility of Big Data. This article provides an in-depth exploration of how HE can be utilized to achieve privacy-preserving analytics in various Big Data applications. By examining different HE schemes, we highlight their relative strengths and weaknesses, focusing on computational efficiency and scalability in the context of Big Data ecosystems. Despite its potential, the implementation of HE for large-scale analytics still faces significant challenges, including high computational overhead and scalability issues. However, ongoing research and development promise to overcome these barriers, making HE an indispensable tool for privacy-preserving data analysis in the future.

References:

- Gentry, C. (2009). "A Fully Homomorphic Encryption Scheme." Proceedings of the 41st Annual ACM Symposium on Theory of Computing.
- Paillier, P. (1999). "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes." EUROCRYPT 1999.

- Gentry, C., & Halevi, S. (2011). "Implementing Gentry's Fully Homomorphic Encryption Scheme." *ACM Transactions on Computational Logic*.
- Shamir, A. (1979). "How to Share a Secret." *Communications of the ACM*.
- Miele, A., & Brackney, M. (2017). "Efficient Privacy-Preserving Big Data Analysis." *IEEE Transactions on Knowledge and Data Engineering*.
- Zhang, W., & Zhang, Z. (2018). "Efficient Privacy-Preserving Big Data Computations Using Homomorphic Encryption." *Journal of Cryptography*.
- Smart, N. P., & Vercauteren, F. (2014). "Fully Homomorphic Encryption with Concrete Efficiency Improvements." *Proceedings of the International Conference on Cryptology*.
- Chabanne, H., & Gascuel, P. (2013). "Homomorphic Encryption for Privacy-Preserving Computations." *IEEE Access*.
- Kim, H., & Wang, X. (2016). "Scalable Homomorphic Encryption Scheme for Big Data Security." *Proceedings of the IEEE International Conference on Big Data*.
- Brakerski, Z., & Vaikuntanathan, V. (2014). "Efficient Fully Homomorphic Encryption from (Standard) LWE." *Proceedings of the IEEE Symposium on Security and Privacy*.
- Nguyen, D. K., & O'Neill, M. (2020). "Challenges in Scaling Homomorphic Encryption for Large-Scale Big Data." *Journal of Big Data*.
- Liu, Y., & Li, Y. (2019). "Practical Implementation of Privacy-Preserving Big Data Analytics Using Homomorphic Encryption." *Proceedings of the International Conference on Data Science and Advanced Analytics*.
- Li, X., & Liu, F. (2020). "Homomorphic Encryption for Secure Machine Learning in Big Data Analytics." *Journal of Information Security and Applications*.
- Zhang, S., & Zhang, T. (2017). "Privacy-Preserving Data Analysis Using Homomorphic Encryption in Cloud Computing." *Journal of Cloud Computing: Advances, Systems, and Applications*.
- Popa, R. A., & Andreeva, G. (2016). "Secure Data Sharing with Homomorphic Encryption in Cloud Environments." *Proceedings of the IEEE Symposium on Security and Privacy*.
- Gentry, C., & Ramzan, Z. (2016). "Computationally Efficient Fully Homomorphic Encryption." *Journal of Cryptographic Engineering*.
- Li, H., & Liu, W. (2018). "Data Privacy in the Internet of Things Using Homomorphic Encryption." *International Journal of Information Security*.
- Shokri, R., & Shmatikov, V. (2017). "Privacy-Preserving Big Data Mining with Homomorphic Encryption." *IEEE Transactions on Privacy and Security*.

- Acar, A., & Neves, R. (2019). "Privacy by Design in Big Data: Homomorphic Encryption for Secure Analytics." Proceedings of the ACM Workshop on Data Privacy.
- Swaminathan, K., & Joshi, A. (2018). "Homomorphic Encryption for Big Data Analytics: A Survey of Challenges and Future Directions." Big Data Research.
- Ahmad, N. R. (2025). *Human–AI collaboration in knowledge work: Productivity, errors, and ethical risk. Journal of Humanities and Social Sciences.*
<https://doi.org/10.52152/6q2p9250>
- Ahmad, N. R. (2025). *Urban water service delivery in emerging economies: Fiscal sustainability, cost recovery, and governance performance. International Journal of Business and Economic Analysis.* <https://doi.org/10.24088/IJBEA-2025-103005>