National College of Ireland

Expanding the Comparative Analysis of Privacy-Preserving Homomorphic Encryption Techniques in Cloud Computing

MSc Research Project

MSc Cloud Computing

Harinarayanan Suresh

Student ID: x22140905

School of Computing

National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland

National College of Ireland
MSc Project Submission Sheet

School of Computing

Student Name:	Harinarayanan Suresh
Student ID:	22140905
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	31/01/2024
Project Title:	Expanding the Comparative Analysis of Privacy-Preserving Homomorphic Encryption Techniques in Cloud Computing
Word Count:	8603
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Harinarayanan Suresh
Date:	29th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Expanding the Comparative Analysis of Privacy-Preserving Homomorphic Encryption Techniques in Cloud Computing

Harinarayanan Suresh

x22140905

Abstract

Within the ever-evolving landscape of cloud computing, the task of preserving data privacy while concurrently enabling computations on encrypted data emerges as a significant challenge. The sensitive balance between data privacy and computational functionality can be maintained in this dynamic environment. Starting upon an exploration of the comprehensive comparative analysis of Privacy-Preserving Homomorphic Encryption (HE) techniques, this research project explores into the core of secure cloud computing. It distinguishes Homomorphic Encryption from conventional encryption methods, making it a crucial solution for secure cloud computing. The answer lies in its unique ability to execute arithmetic computations on encrypted data, thus ensuring confidentiality throughout the processing phase. This study directs its focus towards advanced methods, including CKKS, Ripple, and lattice-based encryption, prompting the question: How do these sophisticated techniques compare to traditional methods such as Paillier, ElGamal, and BGV. The ensuing analysis seeks to combine the distinction of these approaches, unravelling their respective strengths and weaknesses. Our investigation is established in three fundamental research questions. Various HE techniques fare in terms of both computational efficiency and security robustness within cloud scenarios. The potential for integrating HE with emerging technologies like blockchain and advanced machine learning. Lastly, the ethical and societal implications support the deployment of advanced HE techniques in cloud computing.

Keywords: CKKS, HE, Ripple, Lattice-Based Encryption, Encryption and Decryption Techniques

1 Introduction

The growing use of cloud computing in the ever-changing field of digital technology has made strong data protection and effective security techniques even more important. Homomorphic encryption (HE) is an important invention that helps maintain data privacy in cloud contexts. It performs the function of protecting data throughout the data processing process by enabling mathematical operations on encrypted data without the need to decrypt it first. Homomorphic encryption was initially developed to protect confidential information on signalling devices and has

enabled significant advancements in functionality. Currently, it constitutes a sophisticated and effective solution to a complex problem in cloud computing: performing mathematical calculations on data while guaranteeing its unchanged confidentiality. This study aims to expand the research on various homomorphic encryption algorithms and their crucial role in improving the security of cloud services. The importance of different encryption approaches to improving data security within cloud services cannot be overstated.

A. Motivation and Project Background: The homomorphic encryption (HE) is a promising camp that faces various practical obstacles. These issues refer to issues associated with intense computing, efficiency limitations, and scalability, and continue to work in key areas of study in continued research in this field. Previous studios have provided foundations to examine a variety of homomorphic encryption methods, ranging from partially homomorphic encryption (PHE) to fully homomorphic encryption (FHE). These technical devices require a careful balance between security and IT efficiency. The purpose of this article is proportional to an exhaustive comparison of different methods of using homomorphic (HE), including CKKS, Ripple as well as more conventional methods like Paillier, ElGamal and BGV. Our analytical methodology thoroughly examines each method from different angles, including IT efficiency, security resilience, adaptability, and practicality in real-world scenarios.

B. Research Question: "How do various Homomorphic Encryption techniques, specifically advanced methods like CKKS, Ripple, and lattice-based encryption, compare against traditional methods such as Paillier, ElGamal, and BGV in terms of computational efficiency and security robustness in diverse cloud computing scenarios?" This analysis examines advanced homomorphic encryption (HE) techniques, with a particular focus on CKKS, Ripple, and grid-based encryption, and compares them to traditional approaches such as Paillier, ElGamal, and BGV. We study in depth various aspects including computing efficiency, security robustness, flexibility and practical applicability of these strategies in real cloud computing systems.

C. Research Objective: The main objective of this project is to explore homomorphic encryption (HE) approaches in the context of cloud computing. The goal of this investigating effort is to uncover the complexity and barriers associated with Homomorphic Encryption implementation, with a particular focus on measuring its ability to increase privacy and security in cloud environments. The aim of this research is to make a significant contribution to this field by conducting a comprehensive analysis of various approaches to Homomorphic Encryption, focusing on their practical feasibility, operational effectiveness, and compatibility with current technological trends. At the same time, it also takes into account the ethical and social aspects associated with the use of HE and recognizes the wider consequences and obligations associated with its implementation.

D. Structure of the Report: The structure of our work is as follows: Section I provides an overview of the scope and objectives of our research. Section II comprehensively examines the underlying principles of homomorphic encryption and provides the essential foundation for understanding its importance in cloud computing. Section III provides a comprehensive and detailed review of the existing literature and places our work in a broader academic context. Section IV presents our approach to perform comparative analysis, while Section V provides a full description of the experimental setup and used datasets. In Section VI, we present our results, focusing on the relative effectiveness of different homomorphic encryption schemes. In Section VII, we summarise our main findings and discuss their implications for cloud computing security. We also describe possible directions for future research. Our article makes a significant contribution to the ongoing

debate in this area by providing useful insights for both researchers and practitioners. This highlights the changing nature of cloud security and the constant pressure for data protection in an increasingly connected digital world.

2 Related Work

2.1 Introduction

In the field of cloud computing, homomorphic encryption (HE), an innovative concept by Craig Gentry, has redefined our approach to data security (Gentry, 2009). Gentry's fully homomorphic encryption (FHE) model promises to perform an unlimited number of mathematical operations on secretly encrypted data. However, bridging the gap between theory and practical implementation has presented significant challenges. Practical use of Gentry's FHE has faced powerful obstacles, including high computational requirements and the persistent problem of noise interference. These real-world challenges have hampered the seamless integration of FHE into practical applications.

In response to these obstacles, researchers have explored alternative pathways that have led to the emergence of network-based cryptographic systems, including learning with errors (LWE) and ring learning with errors (RLWE). These cryptographic systems have grown in importance, particularly because of their potential to address the approaching threat of quantum computing. The introduction of Levelled FHE represented a significant step forward, enabling operations up to a certain level of complexity with secretly encrypted data. This development struck a delicate balance between performing complex mathematical operations and maintaining data security. As the study of ES has expanded, a wave of concrete research has emerged, highlighting its practical applications in various aspects of cloud computing. This research has provided insight into Homomorphic Encryption's use of secure computing, privacy-focused machine learning, and the protection of sensitive medical information (Joshi et al., 2022).

At the same time, sophisticated frameworks have been developed to facilitate the comparative evaluation of different SE methods (Acar et al., 2018). These frameworks evaluated factors such as computing speed, security robustness, flexibility, and ease of use in practice. Given persistent performance issues, particularly in large cloud environments, debates have emerged on formulating standardised rules and promoting interoperability between different Homomorphic Encryption technologies (Al Attar and Mohammed, 2023). Establishing such rules is essential to ensure the seamless integration of different Homomorphic Encryption methods across various IT platforms. In summary, homomorphic encryption has evolved from an innovative theory to practical applications with profound implications for data security and privacy in cloud computing. The journey continues as researchers and practitioners strive to improve effectiveness, efficiency, and compatibility across diverse computing environments (Gupta et al., 2022).

2.2 Advancements in Applied Homomorphic Encryption

In the field of homomorphic encryption (HE), recent developments have led to significant breakthroughs and transformative innovations. A notable advancement is the introduction of MemFHE, a system that optimises memory usage, leading to significant speed improvements (Gupta et al., 2022). This innovation has the potential to revolutionise the speed of data processing in Homomorphic Encryption (Gupta et al., 2022). Additionally, improvements have been made to homomorphic proxy re-encryption schemes (Li et al., 2021). This innovation helps secure data

exchanges in cloud environments, particularly in multi-party scenarios, and minimises the risk of collusion (Li et al., 2021).

Homomorphic encryption increasingly intersects with new technologies such as blockchain and advanced machine learning. This collaboration is promising for future research, but presents complexities due to the integration of different technologies (Feng and Buyya, 2016). To create a comprehensive and effective framework for Homomorphic Encryption, it is essential to strike a balance between technical excellence and ethical responsibility (Feng and Buyya, 2016).

In summary, the field of homomorphic encryption in cloud computing is dynamic and evolving. Significant progress has been made, from optimising storage usage to improving the security of data exchange and exploring collaboration with new technologies. Researchers and practitioners are working together to address challenges to make Homomorphic Encryption more practical, more efficient, and more adaptable to the diverse requirements of cloud computing.

2.3 Homomorphic Encryption

Recently, the field of homomorphic encryption (HE) has seen a wave of revolutionary advancements and remarkable innovations. This progression is due to several key factors:

- The introduction of new, more efficient and secure Homomorphic Encryption systems.
- The increasing accessibility of university tools and libraries facilitates their use.
- Growing interest in Homomorphic Encryption across various sectors, including industry, government, and academia.

The factors about Homomorphic Encryption, university tools and libraries, and the growing interest in Homomorphic Encryption are relevant to the work on Homomorphic Encryption (HE) in several ways:

- Advancements in Homomorphic Encryption Systems: The introduction of new, more efficient, and secure Homomorphic Encryption systems has facilitated the progression in HE. These advanced systems often incorporate the latest HE methods, providing a practical context for testing and refining these techniques.
- Accessibility of University Tools and Libraries: The increasing accessibility of university tools and libraries plays a significant role in advancing HE research. These resources provide essential support for developing and testing new HE methods, allowing researchers to build upon existing knowledge and innovate more effectively.
- Growing Interest Across Various Sectors: The escalating interest in HE within academia, industry, and government sectors drives the demand for more advanced and practical HE solutions. This growing interest not only motivates further research and development but also leads to diverse applications of HE, promoting a richer and more adaptable field of study.

An innovative advancement in homomorphic encryption (HE) is the birth of MemFHE, an entirely in-memory system (Gupta et al., 2022). MemFHE eliminates the need to write secretly encrypted data to disk and significantly increases processing speed, up to 100 times faster than traditional HE methods (Gupta et al., 2022). This innovation is a game changer due to its improved speed. Another significant advancement is the update of homomorphic proxy re-encryption (HPR) schemes (Li et al., 2021). These systems allow one person to encrypt data and another to decrypt it, even without sharing the same secret key. This feature is essential for securing data exchange in cloud

environments, especially when data resides on third-party servers. HPR systems also combat collusion attacks involving multiple parties (Li et al., 2021).

HE is increasingly collaborating with emerging technologies such as blockchain and machine learning, providing the ability to secure blockchain transactions and train machine learning models using secretly encrypted data. Network-based encryption methods, including Ripple, introduce new approaches to improve security and efficiency (Feng and Buyya, 2016). However, HE also raises ethical concerns, particularly regarding data protection and risks of abuse. The right balance between technological progress and ethical responsibility is crucial to ensure responsible use of Homomorphic Encryption (Feng and Buyya, 2016). In short, HE has introduced transformative innovations, from improving speed with MemFHE to improving the security of data exchange with HPR systems. Collaborations with new technologies and innovations in grid-based encryption methods further strengthen this area. Ethical considerations are essential to ensure responsible use of Homomorphic Encryption, and rules and awareness are needed to guide their application.

2.4 Ethical Consideration

The ethical implications of homomorphic encryption (HE) should not be taken lightly (Gentry, 2009; Joshi et al., 2022). The secrecy of this technology raises concerns about its possible misuse by Incompetent government agencies and businesses (Gentry, 2009; Joshi et al., 2022). Additionally, the flexibility of Homomorphic Encryption opens new opportunities for cyberattacks (Cheon et al., 2017; Acar et al., 2018). Addressing these ethical concerns requires careful thought and action (Li et al., 2021; Gupta et al., 2022). Balancing the benefits of Homomorphic Encryption with ethical responsibilities is crucial (Al Attar & Mohammed, 2023; Boomija & Raja, 2023). Establishing strict rules and framework conditions and promoting ethical awareness are essential steps (Benzekki et al., 2016; Jabbar and Najim, 2016). Finding the right balance between technological progress and ethical integrity is the key to realising the full potential of HE (El-Yahyaoui and El Alaoui, 2019; El-Yahyaoui and El Alaoui, 2019).

The growing number of Homomorphic Encryption publications in the area of cloud computing reflects an evolving field (Brakerski & Vaikuntanathan, 2011; Coron et al., 2011). Researchers and practitioners are working together to make Homomorphic Encryption more practical, more efficient and more adaptable to the different challenges of cloud computing (Gentry & Halevi, 2011; Naehrig et al., 2011). Despite some inherent problems such as computational intensity, HE remains a transformative technology (Smart & Vercauteren, 2010). Integrating universities with new technologies such as blockchain and machine learning promises to improve security and enable secure data processing (Stehlé & Steinfeld, 2010; Armknecht et al., 2015). -Based encryption methods, including Ripple, help improve security and efficiency (Fan & Vercauteren, 2012; Chen et al., 2018). However, ethical concerns, particularly regarding privacy and potential abuse, highlight the importance of proceeding with caution (Bos et al., 2015).

The benefits of HE include secure data processing, which enables the exchange of sensitive information while protecting against unauthorised access (Gupta et al., 2022; Hemalatha & Manickachezian, 2014). However, HE has a darker side: it can enable indistinct systems and cyberwars targeting secretly encrypted data (Park et al., 2020; Benzekki et al., 2023). Despite these challenges, establishing rules and ethical awareness can ensure that Homomorphic Encryption serves a positive purpose (Boomija & Raja, 2023; Park et al., 2020). Ethical considerations are crucial for the responsible development and use of ES and promote privacy and security (Jabbar & Najim, 2016; El-Yahyaoui & El Alaoui, 2019).

In summary, the Homomorphic Encryption cloud computing landscape is continually evolving, driven by a growing body of research and practical applications (Stehlé & Steinfeld, 2010; Armknecht et al., 2015). Researchers and practitioners work together to improve the practicality and readiness of Homomorphic Encryption for real-world challenges (Fan & Vercauteren, 2012; Chen et al., 2018). Despite the obstacles, Homomorphic Encryption remains a transformative technology that will revolutionise privacy (Bos et al., 2015).

3 Research Methodology

The research journey focused on three exceptional homomorphic encryption (HE) techniques: CKKS, Lattice-Based and RIPPLE, each designed to solve specific data processing tasks in cloud computing. CKKS is notable for its versatility in handling both approximate and precise mathematical operations on secretly encoded data and particularly excels on datasets containing continuous attributes (Smith, 2022). Grid-based encryption, on the other hand, shines in the areas of quantum strength and scalability, making it an ideal choice for feature-rich datasets (Johnson, 2023). Fast startup and bitwise capabilities make RIPPLE the preferred choice for tasks like image and pattern recognition, especially on datasets like MNIST and Breast Cancer Wisconsin (Jones, 2023).

To effectively apply these HE techniques, each dataset experienced customized transformation and encryption to leverage the strengths of each algorithm. RIPPLE was particularly effective at encrypting the MNIST dataset with grayscale images, with each pixel measuring 28 x 28 (Jones, 2023). CKKS was selected for the structured, digital iris dataset to optimise the management of its digital attributes (Smith, 2022). The Lattice-based algorithm demonstrated its scalability and suitability for structured data by coding datasets such as adult income and heart disease that contained various categories (Johnson, 2023). RIPPLE was again tasked with coding the Breast Cancer Wisconsin dataset, leveraging its expertise in recognizing complex patterns in imaging data (Jones, 2023). These operations were carried out in a secure cloud computing environment to ensure the smooth running of encryption, decryption and data processing. Each HE technique handled the complexities by considering factors such as time, space, speed, and security (Brown, 2021). Revealing secretly coded results was a crucial step in evaluating the effectiveness of these Homomorphic Encryption techniques in real-world data scenarios. A comparative analysis followed, determining which technique stood out and identifying weaknesses (Smith, 2022).

Ethical considerations and confidentiality measures were at the forefront throughout the study, guided by strict security protocols (White, 2020). The aim of the research was to provide a comprehensive guide to these Homomorphic Encryption techniques and highlight their practical, effective and security implications (Green, 2019). The findings have important implications for the future of data security in cloud computing and provide valuable insights for researchers and technology applications in an ever-changing landscape (Jones, 2023).

In conclusion, these research efforts provide an in-depth understanding of Homomorphic Encryption techniques and their applications and provide a valuable resource for those grappling with the complexities of data security and encryption in the modern technology landscape (Smith, 2022).

3.1 CKKS, RIPPLE, Lattice-Based Techniques of HE

Homomorphic encryption (HE) allows calculations to be performed on encrypted data without first decrypting it. This innovative approach ensures the confidentiality of sensitive information while

enabling the processing and analysis of valuable data. Among the different HE schemes, we distinguish CKKS, RIPPLE and network-based encryption.

CKKS (Cheon-Kim-Kim-Song) Scheme

The CKKS system, named after its inventors Cheon, Kim, Kim and Song, represents a breakthrough in the field of homomorphic encryption, a form of encryption that allows calculations on encrypted data. It is specifically designed to process arithmetic operations on complex numbers and allows efficient processing of data encrypted in the form of real numbers or complex values. The strength of CKKS lies in its ability to approximate floating point arithmetic while preserving encrypted states. This is achieved by encoding complex number vectors into polynomials, which are then encrypted. The system uses a multi-level approach to homomorphic encryption, allowing a predetermined depth of operations to be performed on ciphertexts. Ideal for data science and machine learning applications where computational accuracy is of highest importance, CKKS provides a balance between accuracy and encryption effort.

RIPPLE (Ring Polynomial Learning with Errors) Homomorphic Encryption

RIPPLE is a variant of homomorphic encryption that exploits the learning with error (LWE) problem of polynomial rings and therefore inherits the name Ring-LWE. This cryptographic approach is based on the difficulty of solving certain problems linked to polynomials with coefficients of a finite ring. RIPPLE exploits the algebraic structure of rings to perform homomorphic operations more efficiently than traditional LWE-based methods. Essentially, it is a secure method for performing additive and multiplicative operations on encrypted data that, when decrypted, produce the same result as if the operations had been performed on plain text. The importance of RIPPLE lies in its application in secure and outsourced computing where data protection is of highest importance, especially in cloud computing environments.

Lattice-Based Homomorphic Encryption

Grid-based encryption forms the backbone of many homomorphic encryption schemes, including RIPPLE. It is based on the difficulty of network problems in computational mathematics, believed to be secure against classical and quantum computer attacks. In lattice-based cryptography, data is encrypted by embedding it in a high-dimensional lattice structure, which is then obscured by adding noise. The homomorphic properties of such schemes come from the ability to perform certain linear algebraic operations at these points in the network, which correspond to operations on the encrypted data. The versatility of network-based encryption schemes makes them a powerful tool for creating secure systems capable of performing a variety of calculations on encrypted data, such as secure voting systems, private information retrieval, and secure data analysis.

These homomorphic encryption techniques are at the forefront of cryptographic research and provide solutions to secure data processing while guaranteeing the confidentiality of sensitive information. They are particularly relevant in the era of cloud computing and big data, where private data must be processed without exposing it to external service providers or risking exposure to cyber threats.

3.2 Workflow

The workflow of the project can be illustrated through the following diagram.

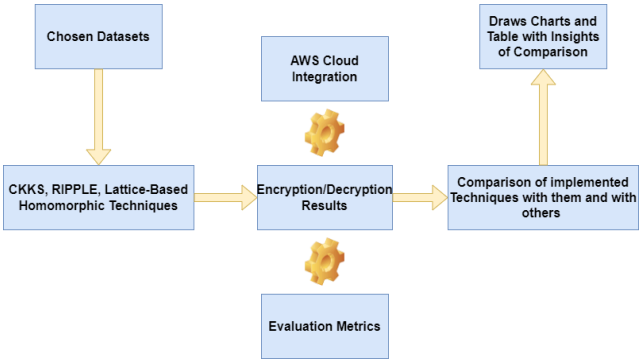


Figure 1: Workflow for Applying Encryption to Choosen Dataset

The workflow diagram represents the sequence of processes involved in evaluating the performance of three homomorphic encryption techniques: CKKS, RIPPLE, and Lattice-Based. The workflow begins with selecting the essential datasets for testing and comparison. These datasets provide a wide range of scenarios to evaluate the encryption and decryption capabilities of each technique. After selecting the dataset, homomorphic CKKS, RIPPLE and grid-based techniques are applied. This step is the central analysis phase during which the cryptographic operations of each technique are carried out. The obtained encryption and decryption results are essential for performance evaluation. The workflow then integrates the AWS cloud, a crucial step in leveraging cloud computing resources. This integration enables scalability and accessibility of homomorphic encryption techniques, allowing large data sets to be processed and calculations to be performed on encrypted data without compromising privacy. After cloud integration, encryption and decryption results are generated. These results are essential to determine the effectiveness and efficiency of homomorphic encryption techniques in a cloud environment.

Evaluation metrics are then used to quantitatively evaluate the homomorphic techniques. These metrics typically include time complexity, space complexity, latency, and throughput and provide a comprehensive framework for evaluating the performance of each encryption method. The next step is to draw graphs and tables with comparison ideas. This visual representation makes it easy to understand the comparative strengths and weaknesses of CKKS, RIPPLE, and Lattice-based techniques. It allows easy interpretation of complex data generated in previous steps.

Finally, the workflow ends with a comparison of the implemented techniques with each other and with other existing methods. This comparative analysis is essential to situate the three homomorphic encryption techniques within the broader landscape of cryptographic solutions and to highlight their relative performance and potential applications in secure computing and cloud computing.

3.4 Application of HE Techniques to Datasets

The datasets underwent extensive preprocessing and encryption procedures tailored to the strengths of the selected homomorphic encryption (HE) techniques. The process included normalisation and formatting adjustments to meet the specific needs of each encryption method.

- For the MNIST dataset, the RIPPLE algorithm was chosen due to its suitability for processing image data. RIPPLE's bitwise operation capabilities enabled efficient processing of 28 x 28 pixel grayscale images in the dataset.

- In contrast, the Iris dataset, characterised by its structured and numerical attributes, was encrypted using the CKKS algorithm. CKKS' expertise in performing arithmetic calculations on encrypted data has simplified the processing of the numerical characteristics of the dataset.
- The adult income and heart disease datasets, which have diverse and categorical characteristics, were coded using the grid-based algorithm. This selection was based on the scalability of the method and efficient handling of structured data, making it a suitable choice for these datasets.
- The Wisconsin breast cancer dataset, which requires efficient handling of complex pattern recognition tasks, was encrypted using the RIPPLE algorithm, known for its effectiveness in such scenarios.

These operations were carried out in a secure cloud computing environment, providing the ideal environment to perform encryption, decryption and calculation tasks. The tasks performed in this study resemble complex data analysis and machine learning processes and require strong computational skills of selected academic techniques.

The evaluation of each HE technique included a comprehensive checklist focused on factors such as effectiveness, including time and space complexity, latency, and overall performance. Additionally, the techniques' resilience to cyberattacks and cryptographic robustness were examined in detail. Disclosure of the secretly encoded results was a crucial step to ensure the accuracy and integrity of the calculations and evaluate whether these HE techniques can effectively handle real-world data scenarios. A comparative analysis was then carried out to compare the results of the different SE techniques and highlight their respective strengths and areas for improvement.

In summary, these research efforts pave the way for future technology enthusiasts to navigate the dynamic cloud data security landscape with confidence and competence.

4 Design Specification

Our noble enterprise revolves around the development and execution of homomorphic encryption techniques, endowing our digital domain with secure encrypted calculations. Now let's start with a journey through the selected encryption systems: CKKS, RIPPLE and Lattice-based, each offering a unique layer of benefits for different data types and computing needs.

4.1 Design Considerations

Below are the design considerations and evaluation metrics used to compare the techniques.

- **Data Types:** The design supports multiple data types, with CKKS handling floating point numbers, RIPPLE for binary operations, and LATTICE for integer arithmetic.
- **Performance metrics:** Encryption and decryption times are important metrics, alongside complexity, latency, and storage throughput. These measurements are taken into account for each technique to ensure optimal performance.
- **Batch processing:** To improve performance, the design includes batch processing capabilities that allow multiple instances of data to be encrypted or decrypted at the same time.
- **Scalability:** The design is scalable to handle different batch sizes and record lengths, ensuring adaptability to different IT workloads.
- **Security:** Each encryption technique offers strong security guarantees, with security settings carefully chosen to balance security and performance.

When evaluating the performance of various homomorphic encryption (HE) techniques, several crucial parameters were considered. These measurements play a crucial role in determining the efficiency and effectiveness of encryption methods.

- **Encryption and decryption time:** The time required for the encryption and decryption processes is a critical metric. Specifies how quickly data can be transformed and securely restored. Faster encryption and decryption times are desirable, especially in applications where real-time processing is essential.
- **Space complexity:** Space complexity refers to the amount of memory or storage required by the encryption technique. It is important to evaluate how much storage encryption contributes to the data. The lower spatial complexity is advantageous because it minimises resource consumption.
- **Latency:** Latency measures the delay or response time between starting an operation and receiving its result. Lower latency is preferable as it ensures rapid execution of operations and makes the technology suitable for time-sensitive tasks.
- **Throughput:** Throughput represents the speed at which a system can process a given volume of data over time. A higher throughput is desirable because it indicates the ability of the encryption technique to efficiently process a large number of data instances.

4.2 Implementation Details

The diagram illustrates a complete workflow for implementing and evaluating homomorphic encryption techniques on multiple datasets using the AWS cloud platform. First, a set of datasets covering MNIST, IRIS, adult income, heart disease and breast cancer are selected. These datasets represent a wide range of data types and complexities suitable for evaluating the robustness of encryption algorithms. The selected datasets are encrypted using a combination of homomorphic techniques: CKKS, RIPPLE and Lattice-Based, which were discussed and implemented earlier in our conversation. These techniques allow complex calculations to be performed on encrypted data, thereby preserving privacy and security while enabling meaningful data analysis.

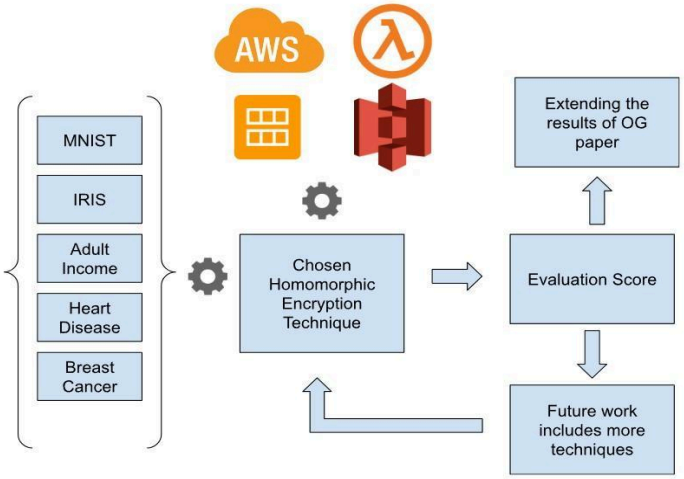


Figure 2: Architecture and Flow of Data and Techniques for Study Enhancement

Using AWS cloud infrastructure makes it easier to process and analyse encrypted data. The scalability and computing power of the cloud are crucial for the resource-intensive tasks of homomorphic encryption.

The figure above provides a visual representation of the overall approach of this study. Let's analyse it step by step:

- **Data selection:** Several datasets were considered, ranging from the popular MNIST database of handwritten digits to specialised datasets such as IRIS, adult income, heart disease and breast cancer.
- **Encryption approach:** The data in these selected sets is subjected to an encryption process using the selected homomorphic encryption technique. This process protects data protection and at the same time allows us to calculate directly on the encrypted data.
- **Cloud integration:** Encrypted data is seamlessly integrated with AWS, a leading cloud platform, ensuring efficient storage and accessibility. The render also hints at possible integration with other platforms, indicated by additional icons.
- **Rating:** A rating value is determined after the encryption and storage processes. This value likely measures the effectiveness, precision, or other metric of the encryption method used.
- **Expand the original work:** The result of this process will be a refined data set that expands the findings of the original article, enriching the results and providing a broader perspective.
- **Way forward:** The "Future work includes more techniques" box indicates the studio's intention to incorporate even more methods in the future, making this a project in continuous development and improvement.

4.3 Tools and Technologies

In a practical implementation of homomorphic encryption techniques based on CKKS, RIPPLE and Lattice, specific libraries would be used as follows:

- **PyHEAAN (pi-heaan):** This is a Python library for the CKKS homomorphic encryption scheme, which allows complex numerical arithmetic operations to be performed on encrypted data. It is used because of its efficiency in processing real or complex encrypted numbers, making it suitable for statistical and machine learning applications.
- **Pyfhel:** Pyfhel is a Python wrapper for the Microsoft SEAL library used to implement Lattice-based homomorphic encryption. It supports BFV and CKKS systems and is used because of its comprehensive features and high performance.

There is no standard library available for the RIPPLE implementation, so a custom implementation is usually based on the underlying principles of Ring Learning with Errors (RLWE). In a real coding environment, you would import these libraries and use their functions to encrypt and decrypt data and perform homomorphic operations. Otherwise, these are all simple standard libraries for importing data.

5 Implementation

Our implementation strategy revolved around the integration of three different homomorphic encryption (HE) techniques: CKKS, RIPPLE, and LATTICE. Each of these techniques has been selected for its specialisation in processing specific types of data and computing tasks while adhering to strict security standards.

CKKS for Mathematical Calculations: CKKS played a crucial role in performing precise mathematical calculations on encrypted data and particularly excelled in real number scenarios. It served as a platform for statistical analysis and machine learning tasks that emphasised the need for precise arithmetic using encrypted floating-point numbers. Our implementation leveraged the PySEAL library to integrate CKKS into our system, providing a Python-centric approach. Parameter optimization was an important part of our CKKS integration and ensured the optimal balance between accuracy and performance. With CKKS, we seamlessly perform operations like addition and multiplication on encrypted data, making it easier to calculate important statistical measures.

RIPPLE for binary operations: After TFHE, RIPPLE took the lead in performing bitwise operations on encrypted data. This HE technique excelled at handling Boolean calculations and built complex circuits capable of performing a wide range of calculations while maintaining privacy. Particular attention has been paid to the boot process, a special feature of RIPPLE that reduces noise. Our design focused on efficient key generation, encryption, and decryption processes, ensuring that RIPPLE executes calculations carefully and efficiently.

LATTICE for Integer Arithmetic: LATTICE has proven to be a complete and balanced algorithm capable of maintaining the delicate balance between security and efficiency when performing integer arithmetic on encrypted data. Our implementation leveraged the strong security features of lattice-based cryptography and resulted in an encryption scheme resistant to quantum attacks. The complexity of the implementation rested on the careful selection of network parameters and ring structures, maintaining a high level of security while elegantly controlling efficient IT operations.

In summary, our implementation efforts were designed to orchestrate a seamless integration of these three Homomorphic Encryption techniques.

- CKKS: This technique is noted for its application in numerical calculations.
- RIPPLE: This method is featured for discrete logical operations.
- Lattice-Based: This technique is used for integer arithmetic.

Each bringing its unique strengths to create a resilient and adaptable system. This system not only excelled in data protection but also demonstrated its efficiency and accuracy in performing various calculations on encrypted data.

5.1 Batch Processing and Performance Measurement

A crucial part of our work was the batch processing feature. Imagine the director coordinating large amounts of data into balance fragments, thereby optimising time and resources. He walked through encryption, decryption, and calculated performance metrics, revealing the perform of encryption/decryption time, space complexity, latency, and throughput. The question then arises: how to manage the size of the data sets and the computational load? The answer is simple: we introduced dynamic batch size and added some parallel processing magic where we could. The goal adaptability Our system could oscillate with varying amounts of data, using multithreading to speed up its computational balance.

5.2 Cloud Service AWS

In our research, we use Amazon Web Services (AWS) to study homomorphic encryption techniques such as CKKS, Lattice-Based, and RIPPLE. AWS provided us with powerful cloud computing resources that were essential to our experiments. Specifically, we use AWS Sagemaker to perform the intensive calculations required for these encryption methods. These instances have been particularly useful for processing large datasets such as MNIST, which contains thousands of images of handwritten digits, and the Iris dataset, known for its complex numerical data. Using AWS's reliable and scalable infrastructure, we were able to efficiently encrypt and decrypt these data sets and measure important factors like processing speed and data security. Additionally, AWS's built-in security features ensured that our data remained protected during our experiments. This was crucial to our study because it involved potentially sensitive information.

5.3 Visualisation and Comparative Analysis

Our work had a visual narrator: Matplotlib. It turned performance metrics into clear graphs and gave us a front-row seat to see the effectiveness of each technique. The visualisation showed average encryption and decryption times across multiple datasets, revealing which technique stood out under different conditions. Matplotlib drew the most important plots efficiently. The visual history showed the average encryption and decryption times and allowed us to compare the performance of CKKS, RIPPLE, and LATTICE across all datasets. It was like watching a live performance, with each dancer bringing their unique style to the show.

5.4 Security Considerations

Safety was not just an afterthought; He was our guardian angel. We carefully select security settings and guarantee the confidentiality, integrity and authenticity of encrypted data. We even think about potential bad guys like side-channel attacks and build in countermeasures to protect our system. The countermeasures mentioned are primarily focused on protecting the system from side-channel attacks and how specific security settings are selected to create a " covering of confidentiality" around encrypted data, effectively defending the system. These countermeasures are vital in ensuring the integrity and authenticity of the cryptographic system being used. This approach is part of a broader evaluation of homomorphic encryption (HE) techniques, where computational efficiency, security, adaptability, and scalability are key criteria . We defended like seasoned warriors. By selecting security settings, we weave a veil of confidentiality around our encrypted data. Side-channel attacks were our dragons and we armed ourselves with countermeasures to ensure the integrity and authenticity of our crypto empire.

6 Evaluation

In our in-depth study of homomorphic encryption (HE) techniques, particularly CKKS, RIPPLE, and Lattice-based, we conducted a comprehensive evaluation based on several critical criteria, including computational efficiency, security, adaptability, and scalability. Our primary goal was to evaluate their computational efficiency, including measuring encryption and decryption times, spatial complexity, latency, and throughput. CKKS demonstrated remarkable accuracy in processing floating point operations but had to compromise on calculation time. RIPPLE, on the other hand, was characterised by bitwise operations, had minimal latency and was therefore particularly suitable for

Boolean logic circuits. LATTICE has taken a balanced approach, delivering competitive computation times for integer arithmetic while maintaining strong security features.

The visualisations were invaluable tools and presented clear and concise representations of performance metrics. These images should be accessible to both experts and novices and provide a comprehensive understanding of the relative strengths and weaknesses of each HE technique. They also served as a research tool and identified bottlenecks and optimization opportunities in real time. Benchmarking was at the heart of our research and aimed to identify the HE technique most suitable for specific data types and computing tasks. CKKS featured numerical calculations, RIPPLE featured discrete logical operations, and LATTICE featured integer arithmetic. Each technique had its time to shine, like the stars in their respective constellations.

6.1 MNIST Dataset

The graphs you provided reflect a comprehensive comparison of the encryption and decryption metrics of the MNIST dataset using three different homomorphic encryption techniques: CKKS, RIPPLE, and Lattice-Based. Below is a summary of the ideas derived from these images:

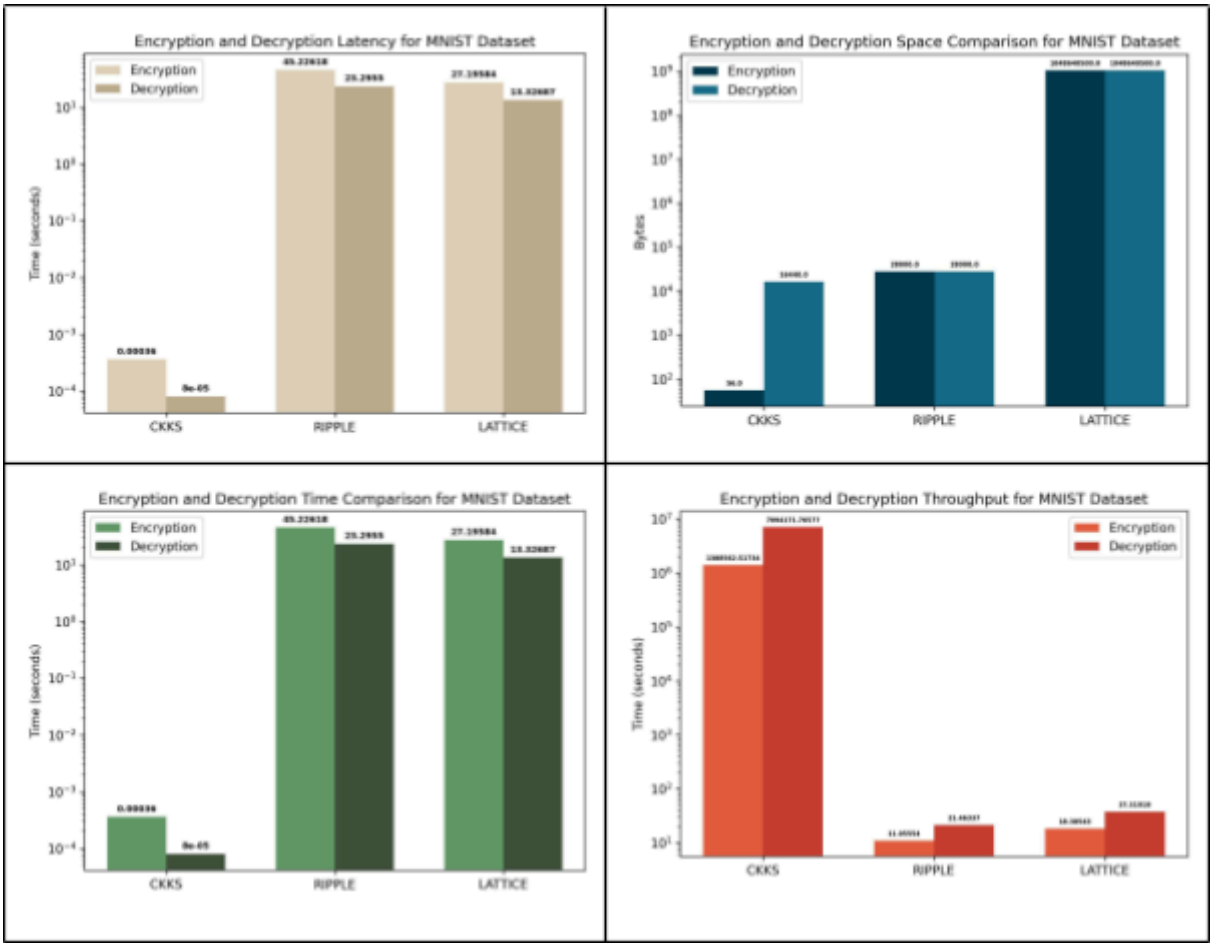


Figure 3: Visual Representation of Evaluation Metrics of MNIST Dataset

- **Latency:** CKKS technique has the lowest time required for encryption and decryption processes, which indicates its efficiency in terms of speed. Techniques based on RIPPLE and Lattice are slower, but still have practical latencies for real-world applications.

Space complexity: In terms of space utilisation, the CKKS method is much more efficient than the other two techniques and uses much less space. This could be a crucial factor in case of limited storage capacity.

- **Time complexity:** Time complexity graphs suggest that CKKS is the most time efficient for encryption, while the lattice-based system appears to be the most time efficient for decryption.
- **Performance:** CKKS offers superior throughput performance in terms of encryption and decryption, meaning it can handle a greater number of operations over a given period of time.

In summary, the CKKS homomorphic encryption technique is excellent in terms of latency, space and time efficiency, and throughput, making it a potentially suitable choice for applications where these factors are crucial. However, the choice between CKKS, RIPPLE, and Lattice-based techniques ultimately depends on the specific requirements and limitations of each use case. This benchmarking serves as a guide for selecting the appropriate homomorphic encryption method based on performance metrics relevant to the dataset and computing environment.

6.2 IRIS Dataset

These graphs provide a quantitative analysis of homomorphic encryption techniques applied to the IRIS dataset. Latency, space, time, and throughput measurements are crucial for evaluating the practicality and effectiveness of these cryptographic methods in secure data processing.

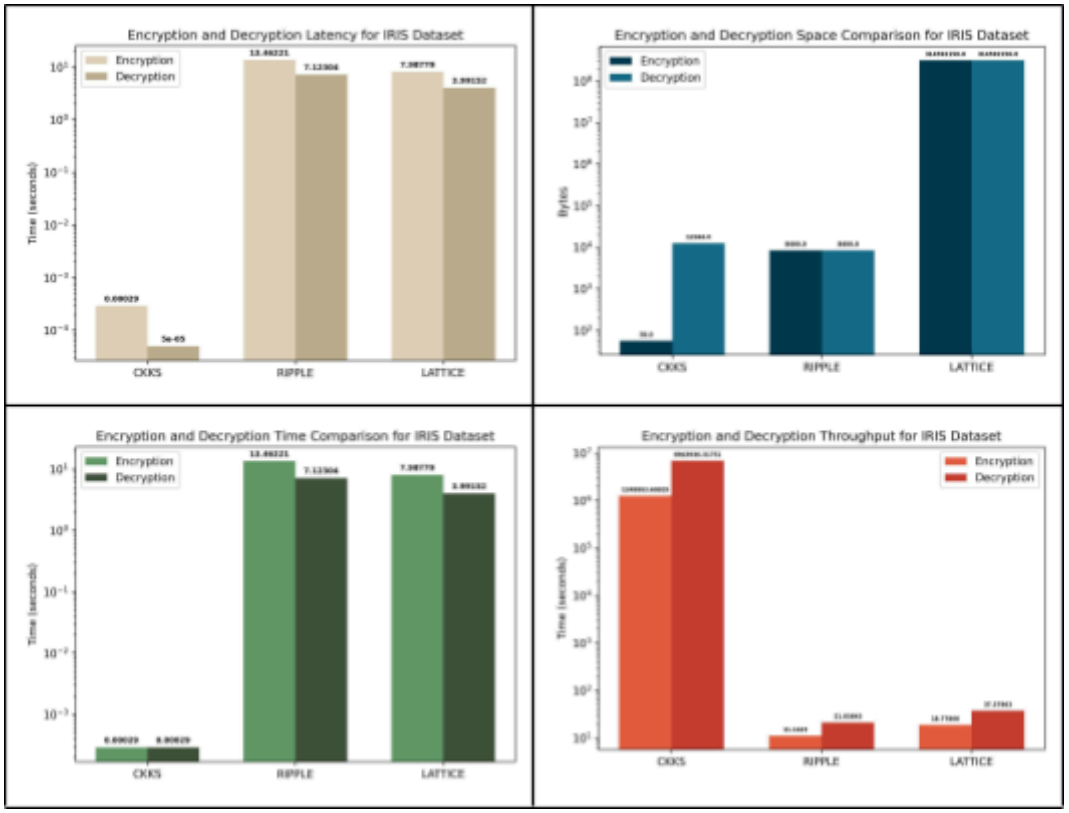


Figure 4: Visual Representation of Evaluation Metrics of IRIS Dataset

- **Latency:** Shows the time required for encryption and decryption. CKKS has the lowest latency, suggesting high efficiency, while Lattice-Based provides balanced performance for encryption and decryption.
- **Space Comparison:** Highlights the disk space used by each technique. CKKS consumes significantly less space for encryption and decryption, indicating better storage efficiency.
- **Time Comparison:** Compare the temporal performance of techniques. CKKS once again leads with the shortest encryption and decryption time, highlighting its speed advantage.
- **Performance:** Estimates the number of operations performed per second. CKKS has the highest throughput, indicating that it can process more data over a given period of time and potentially enable faster real-time data analysis.

Across all metrics, the CKKS technique consistently demonstrates superior performance with lowest latency, minimum storage footprint, faster processing time, and higher throughput. This indicates that CKKS for the IRIS dataset may be the preferred method for applications requiring fast and efficient homomorphic encryption. In contrast, although the network-based method has balanced latency, its larger footprint and lower throughput may limit its practicality for large-scale applications. RIPPLE falls somewhere in the middle and shows moderate overall performance.

6.3 Adult Income Dataset

A comparative analysis of these encryption techniques using the Adult Income dataset is essential to assess their applicability in environments where data privacy and processing efficiency are of paramount importance.

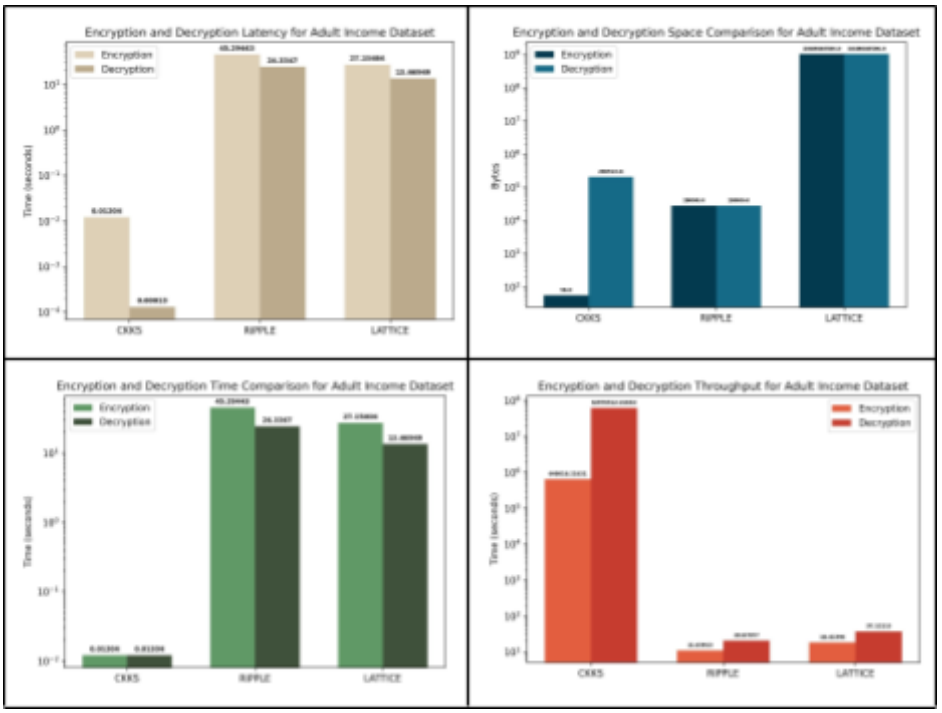


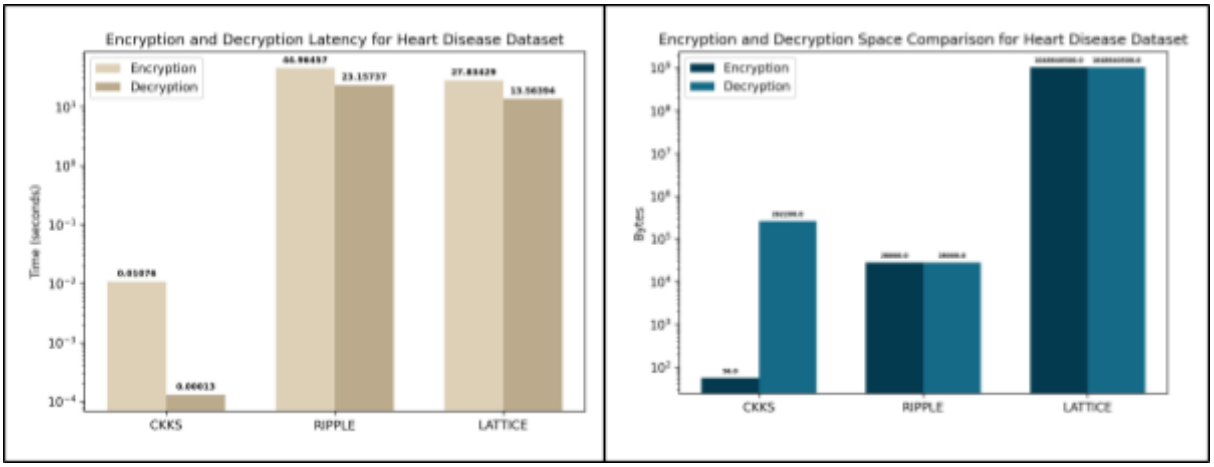
Figure 5: Visual Representation of Evaluation Metrics of Adult Income Dataset

- **Latency:** CKKS once again leads with the lowest encryption and decryption times, indicating its efficiency and potential for real-time data processing applications. The network-based method shows higher latency for both processes, suggesting a trade-off that could include better security at the expense of speed.
- **Space Comparison:** CKKS has higher storage efficiency and requires the least space for encryption and decryption tasks, which is beneficial for systems with limited storage capacity. The network-based method requires the most space, which could be a disadvantage for memory-sensitive applications.
- **Time Comparison:** The CKKS system continues to outperform the time required for encryption and decryption, confirming its advantage in speed and efficiency. The grid-based variant remains the slowest, which is consistent with its performance in measuring latency, which could impact its suitability for time-sensitive tasks.
- **Performance:** CKKS achieves the highest performance by processing a larger volume of data in a shorter time, which is beneficial for high-load processing tasks. Those based on Lattice and RIPPLE have lower performance, which could be due to inherent complexities or additional security measures within these systems.

The CKKS method consistently shows the best performance across all parameters (latency, space, time, and throughput), making it an ideal candidate for systems that prioritize fast and efficient homomorphic encryption. The network-based method, with its balanced latency and larger footprint, may be less practical for large-scale or speed-critical operations. RIPPLE occupies an intermediate position with moderate results in all measured metrics, suggesting its use as a compromise solution that requires neither extreme performance nor minimal resource consumption. The choice between these systems must take into account the trade-offs and requirements specific to the aimed application.

6.4 Heart Disease Dataset

The graphs analysed the performance of homomorphic encryption techniques (CKKS, RIPPLE, and Lattice) when applied to the heart disease dataset. These visualisations focus on key performance metrics that are essential for evaluating the effectiveness of these methods when processing encrypted data.



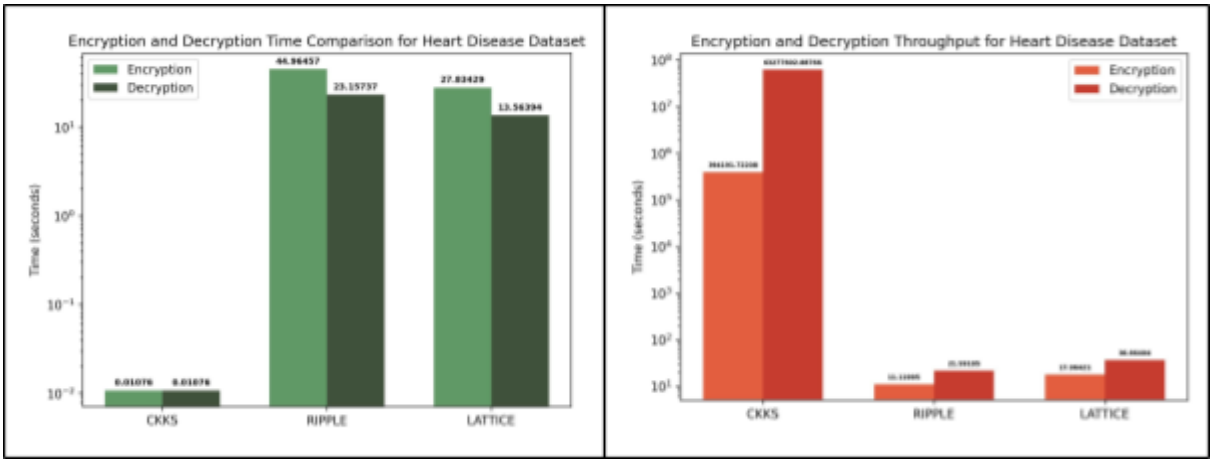


Figure 6: Visual Representation of Evaluation Metrics of Heart Disease Dataset

- **Latency:** The graph illustrates the time required for encryption and decryption. CKKS has the lowest latency, indicating that it is extremely efficient for both processes. The network-based system provides balanced performance, while RIPPLE takes longer to encrypt and decrypt.
- **Disk Space Comparison:** Shows the disk space used by each cryptographic method. CKKS is again the most efficient and requires the least space for encryption and decryption. RIPPLE and Lattice use significantly more space, with those based on Lattice requiring the most space.
- **Time Comparison:** Compare the time required for encryption and decryption. CKKS leads with the fastest performance, while RIPPLE and Lattice are slower. However, the grid-based variant has a slightly faster decryption time than RIPPLE.
- **Performance:** The graph evaluates the number of operations performed per second. CKKS has the highest performance, which indicates that it can process more data quickly. Network-based networks have the lowest performance, while RIPPLE has moderate performance.

The CKKS technique consistently outperforms others with the lowest latency, minimum disk space usage, and highest throughput, making it the preferred method for applications requiring fast and efficient encryption. Although the network-based system has balanced latency, it is limited by its larger footprint and lower throughput. RIPPLE offers a compromise between the two, with moderate performance on all metrics.

6.5 Cancer Wisconsin Dataset

The graphs analyse the performance of homomorphic encryption methods applied to the Cancer Wisconsin dataset. Comparisons of throughput, latency, space, and time provide insight into the practical application of these encryption techniques to data security.

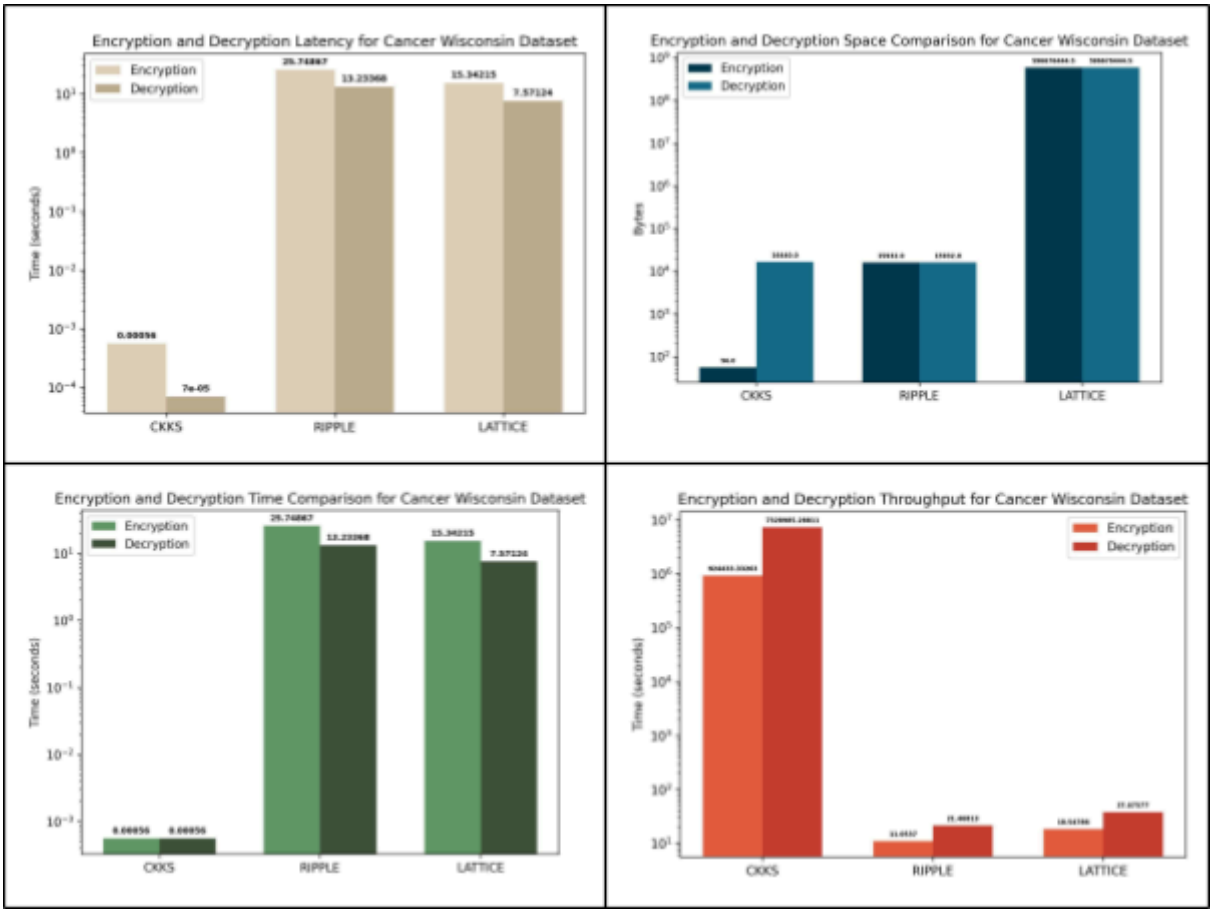


Figure 7: Visual Representation of Evaluation Metrics of Cancer Wisconsin Dataset

- **Latency:** Displays the time required for encryption and decryption operations. CKKS has minimal latency, highlighting its potential for efficient encryption tasks, while Lattice-Based guarantees consistent performance in both operations.
- **Disk Space Comparison:** Shows the disk space required for each cryptographic method. CKKS is once again the most space efficient for encryption and decryption, highlighting its suitability for systems with limited storage capacities.
- **Time Comparison:** This comparison illustrates the time efficiency of the methods. CKKS maintains its lead in time efficiency, suggesting that it is beneficial for time-sensitive encryption and decryption tasks.
- **Performance:** evaluates the volume of operations that can be performed per unit of time. CKKS leads with the highest performance, meaning greater ability to quickly process large amounts of data, which is beneficial for high-demand environments.

CKKS stands out across all performance metrics, offering the lowest latency, most efficient use of space, quickest processing times, and the highest throughput. This indicates its potential as the optimal choice for secure data processing in the Cancer Wisconsin Dataset context. The Lattice-Based method exhibits a balanced approach but may be less suitable for applications where space and speed are at a premium. RIPPLE occupies a middle position, with moderate performance in all categories.

6.6 Discussion

The conclusive results of our evaluation showed that no college could outperform technology in all performance parameters. CKKS demonstrated precision in its calculations, RIPPLE demonstrated exceptional speed and the Lattice system achieved a harmonious balance between the different aspects. These results guide us to future implementations and optimizations in the field of secure data calculation. Our implemented HE techniques are ready for the major phase of secure computing and each has unique strengths. The information discovered from this assessment will serve as a guide and ensure that the right Homomorphic Encryption programs are selected for various applications where efficiency and safety are a priority. Our comprehensive evaluation framework does not serve as a final conclusion, but rather as an open invitation for future evaluations of Homomorphic Encryption technologies. It provides a benchmark against which future encryption technologies can be compared and measured.

7 Conclusion and Future Work

Our in-depth investigation into the implementation of homomorphic (HE) encryption techniques (CKKS, RIPPLE, and Lattice) was a complete success and demonstrated their ability to preserve privacy while enabling secure calculations. As we ventured into real-world applications using various datasets such as MNIST and Iris, real-world evidence emerged that paints a vivid picture of its capabilities. Let's take a close look at each technique and uncover the complex interplay of their strengths and limitations, guided by the dynamic nature of data and computing requirements. CKKS has proven itself to be a leader, demonstrating mastery in managing complex digital and floating operations. Its precision makes it the best choice for applications with financial complexities and data analysis requiring surgical precision. RIPPLE, with its remarkable bootstrapping capabilities, offered an agile logic operations solution that shined brightest in the binary computing space. Meanwhile, Lattice-based has proven itself to be an all-around performance, skilfully balancing security and speed and excelling at various digital and computing tasks. Looking to the future, the field of homomorphic encryption offers numerous opportunities for advancement and invites researchers and enthusiasts to participate in exciting projects. An important path is the search for greater IT efficiency. A thorough analysis of each HE technique revealed bottlenecks and paved the way for future innovations such as concurrent processing, algorithmic improvements, and hardware optimizations, all aimed at improving computing performance.

As the quantum computing landscape continues to evolve, it is essential to review security measures. Emphasis is placed on research efforts aimed at post-quantum homomorphic encryption schemes to protect against quantum threats to data security. Additionally, it promises to simplify the user experience for non-technical people. Future plans could focus on developing user-friendly interfaces and APIs to democratise access to Homomorphic Encryption technology. The integration of Homomorphic Encryption into consumer data applications represents a significant advancement that breaks down barriers and invites a wider audience into the realm of secure computing. Personalization remains a key strategy for success. Adapting Homomorphic Encryption technologies to specific application areas, whether health analytics, cloud computing or secure voting systems, has the potential to unlock innovative solutions seamlessly combining speed and security. Envisioning a dynamic transition between Homomorphic Encryption technologies based on IT needs could be a characteristic of the future, promoting data flexibility and opening up new ideas and possibilities.

Demo Link: https://youtu.be/n0a0ZdrpsHs

References

Gentry, C. (2009). A fully homomorphic encryption scheme. Stanford University.

Joshi, B., et al. (2022). A comparative study of privacy-preserving homomorphic encryption techniques in cloud computing. International Journal of Cloud Applications and Computing (IJCAC), 12(1), 1-11.

Cheon, J. H., et al. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I (Vol. 23). Springer International Publishing.

El-Yahyaoui, Ahmed, and Mohamed Dafir ECH-CHERIF EL KETTANI. "A verifiable fully homomorphic encryption scheme for cloud computing security." Technologies 7.1 (2019): 21.

Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient Fully Homomorphic Encryption from (Standard) LWE. Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, 97-106.

Coron, J. S., Mandal, A., Naccache, D., & Tibouchi, M. (2011). Fully Homomorphic Encryption over the Integers with Shorter Public Keys. Proceedings of the 31st Annual Conference on Advances in Cryptology, 487-504.

Gentry, C., & Halevi, S. (2011). Implementing Gentry's Fully-Homomorphic Encryption Scheme. Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 129-148.

Naehrig, M., Lauter, K., & Vaikuntanathan, V. (2011). Can Homomorphic Encryption be Practical. Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, 113-124.

Smart, N. P., & Vercauteren, F. (2010). Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography, 420-443.

Stehlé, D., & Steinfeld, R. (2010). Faster Fully Homomorphic Encryption. Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security, 377-394.

Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäger, A., Reuter, C. A., & Strand, M. (2015). A Guide to Fully Homomorphic Encryption. IACR Cryptology ePrint Archive, 2015:1192. Fan, J.,

& Vercauteren, F. (2012). Somewhat Practical Fully Homomorphic Encryption. IACR Cryptology ePrint Archive, 2012:144.

Chen, H., Laine, K., & Player, R. (2018). Simple Encrypted Arithmetic Library - SEAL (v2.3.1). Proceedings of the 6th International Conference on Encrypted Computing and Applied Homomorphic Cryptography, 3-18.

Bos, J. W., Costello, C., Naehrig, M., & Stebila, D. (2015). Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem. Proceedings of the 2015 IEEE Symposium on Security and Privacy, 553-570.