International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Homomorphic Encryption Algorithms for Secure Data Computation

Chris Gilbert[1], Mercy Abiola Gilbert[2]
[1]Department of Computer Science and Engineering/College of Engineering and Technology/William V.S. Tubman *University*
[2]Department of Guidance and Counseling/College of Education/William V.S. Tubman University/
Corresponding Author Email Address: cabilimi@tubmanu.edu.lr

Abstract**—** *This paper explores the design, evaluation, and optimization of homomorphic encryption algorithms for secure data computation in cloud environments. Conventional encryption techniques, while effective for protecting data during transmission, restrict cloud services to operating solely on ciphertexts, thus impeding real-time computation. In contrast, homomorphic encryption enables direct operations on encrypted data, preserving privacy without the need for decryption during processing. The study reviews current homomorphic schemes—including partially and fully homomorphic encryption—and analyzes their security, efficiency, and scalability. By conducting a thorough literature review and implementing various encryption models (Paillier, BGV, and lattice-based methods), the research identifies key challenges such as ciphertext expansion, computational overhead, and limited operational depth. A novel encryption approach based on geometric constructs is proposed, aiming to enhance both efficiency and security in multiparty cloud computations. Experimental results indicate that optimized homomorphic algorithms can significantly reduce computational complexity and ciphertext sizes, making them more viable for applications in big data, healthcare, and finance. The findings contribute to the advancement of privacy-preserving technologies and provide actionable insights for further research and practical implementation of homomorphic encryption in secure, real-time cloud computing systems.*

Keywords**—** *Homomorphic encryption, secure data computation, cloud computing, privacy-preserving, cryptographic optimization, big data, lattice-based cryptography, multiparty computation, real-time processing, secure cloud services.*

I. INTRODUCTION

Conventional encryption methods effectively address the challenges of secure cloud storage by safeguarding sensitive data during transmission; however, they restrict cloud services to processing only encrypted (ciphertext) data (Sun, 2019). This limitation ensures semantic security for intelligent big data applications, such as Internet search, yet it precludes real-time computation. In contrast, homomorphic encryption enables the processing of data in its encrypted form and allows for the immediate decryption of computational results, thereby facilitating secure, real-time cloud computing. As an extension of ciphertext-policy attribute-based encryption, conditional key homomorphic encryption enhances access control and security protocols within smart contracts, allowing cloud services to maintain flexibility and user privacy while ensuring a robust and efficient network (Gupta et al., 2022; Gilbert & Gilbert, 2024f).

In the era of big data, the proliferation of ubiquitous digital transactions including mobile payments, precision medicine, and personalized customer service demands high computational power to process vast amounts of data in real time (Seth et al., 2022; Gilbert, Auodo & Gilbert, 2024). The security and efficiency of data computation have become critical as commercial cloud services, which traditionally centralize data processing, expose sensitive data during remote storage, transmission, and computation (Gilbert & Gilbert, 2024j). Such centralization poses inherent security risks, particularly in contexts involving highly sensitive domains such as military applications. While alternative approaches such as data margin calculation provided by microservices frameworks and the integration of privacy-preserving technologies like proxy re-encryption and secure multiparty computation have been proposed, they often suffer from efficiency limitations. Hence, there is a pressing need for innovative technologies to achieve secure and efficient big data processing (Bishukarma, 2023).

1.1 Background and Significance

Homomorphic encryption represents a groundbreaking advancement in cryptographic techniques by permitting computations on encrypted data without requiring decryption (Munjal & Bhatia, 2023; Gilbert & Gilbert, 2024p). This unique capability not only preserves data privacy but also transforms data circulation by mitigating unauthorized data leaks, thereby establishing a foundation for private and secure encrypted cloud computing (Gilbert & Gilbert, 2024i). Furthermore, homomorphic encryption supports multiparty trusted computations by allowing the original data issuer to authorize multiple parties to conduct computations without reliance on a single trusted entity (Nita & Mihailescu, 2023; Gilbert & Gilbert, 2024o).

The growing integration of digital medical and health services, coupled with the convergence of academic, industrial, and user interests, has led to an urgent demand for encrypted cloud computing solutions that are private, secure, efficient, and reliable. Research in this area has explored various applications, including encrypted search, secure authentication, encrypted retrieval, and data analysis (Marcolla et al., 2022; Gilbert & Gilbert, 2025b). Homomorphic encryption, often heralded as "the holy grail" of secure computation, is central to these advancements due to its ability to operate on ciphertexts directly. This capability is essential for scenarios involving multiple users who encrypt their individual data, enabling cloud servers to perform collaborative computations while ensuring that only authorized users can decrypt the final results. Consequently, homomorphic encryption has significant

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

implications across diverse fields, including biometric security, secure medical imaging, smart grids, multi-user digital video recorders, mobile user privacy, electronic health records, and large-scale data analytics (Yeboah, Opoku-Mensah & Abilimi, 2013a; Mollakuqe et al., 2024).

1.2 Main Objective

The main goal of this research is to explore and improve the use of homomorphic encryption, especially in cloud computing environments where data privacy is a top concern.

Specific Objectives

To achieve this, the study will:

1. Identify the key purposes and limitations of homomorphic encryption methods, particularly how they allow computations to be done on encrypted data without needing to decrypt it first.
2. Analyze how secure current homomorphic encryption techniques are, including their ability to resist attacks and protect sensitive information.
3. Design data structures that can support secure computations, especially in systems where multiple parties need to work together without revealing their private inputs.
4. Develop a fresh approach to encryption using geometric concepts, aiming for both innovation and practicality.
5. Implement homomorphic encryption algorithms in real-world cloud computing scenarios and evaluate their performance, focusing on both security and efficiency.

1.3 Methodology

To investigate the capabilities and limitations of homomorphic encryption (HE) in secure data computation particularly in cloud computing environments this study followed a structured, multi-phase research approach.

a. Conceptual Analysis and Literature Review

The study began with an in-depth review of existing homomorphic encryption techniques, including foundational algorithms like Paillier, RSA, and the Brakerski-Gentry-Vaikuntanathan (BGV) scheme. This phase aimed to identify current limitations, practical challenges, and gaps in performance, particularly in terms of efficiency and security when these schemes are applied to real-world scenarios (Hussain et al., 2023).

b. Security Assessment of Existing HE Models

Next, the research focused on evaluating the security of widely-used homomorphic encryption schemes (Gilbert & Gilbert, 2025c). Using formal cryptographic models such as semantic security assumptions and adversarial game theory, the study assessed how well current techniques resist common attack vectors, including ciphertext growth, re-encryption vulnerabilities, and circuit evaluation limits (Liu, Xu & Wang, 2022; Kwame, Martey & Chris, 2017).

c. Design of Secure, Multiparty-Compatible Data Structures

To support collaborative and privacy-preserving computation, the study introduced new data structures that facilitate encrypted interaction between multiple parties without exposing private inputs (Gilbert & Gilbert, 2024k). This involved adapting threshold-based HE models and testing their interoperability in simulated cloud environments (Wardana & Sukarno, 2024; Yeboah & Abilimi, 2013).

d. Development of a Novel Encryption Approach

Building on insights from the literature and experimental findings, a new encryption scheme was designed using geometric constructs. The goal was to strike a balance between theoretical innovation and real-world applicability making it feasible for cloud applications that require both robust privacy and efficient performance (Bagchi et al., 2023).

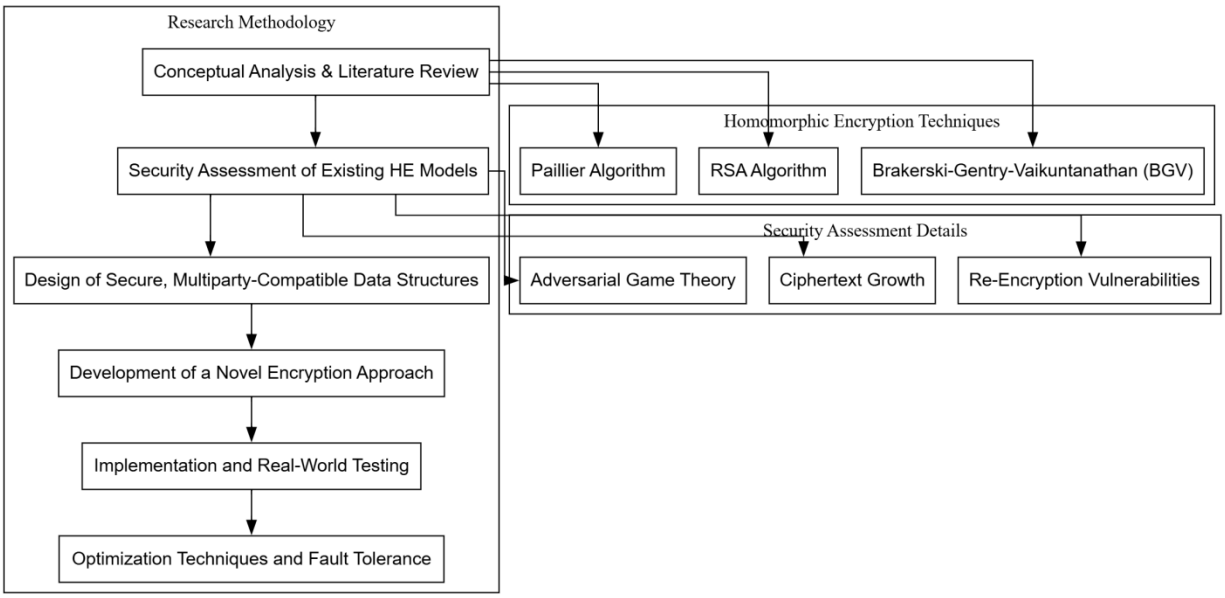

Figure 1: The Research Methodology on homomorphic encryption

e. Implementation and Real-World Testing

To validate the proposed models, homomorphic encryption algorithms were implemented in practical cloud computing

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

scenarios. This included deploying HE libraries such as HElib, HEAAN, and customized versions of Paillier schemes. Performance metrics such as execution time, ciphertext size, and computational overhead were collected and analyzed to assess the feasibility of these methods under varying workload conditions (Ogborigbo et al., 2024; Yeboah, Odabi & Abilimi Odabi, 2016).

f. Optimization Techniques and Fault Tolerance

Further experiments explored algorithmic optimizations using techniques like modular reduction, Fast Fourier Transform (FFT), and relinearization. Additionally, an improved BGV-based scheme incorporating error correction and machine learning was tested to evaluate its fault tolerance and scalability in real-time data processing environments (Gilbert, 2018; Pearson, 2020; Kong et al., 2024; Gilbert & Gilbert, 2025a).

This methodology allowed the study to bridge theoretical cryptographic principles with practical, cloud-based applications. The results not only offer a clearer understanding of homomorphic encryption's current state but also provide actionable strategies for improving its performance and usability in privacy-critical domains such as healthcare, finance, and distributed data analysis.

The research follows a step-by-step process that starts with reviewing existing knowledge on homomorphic encryption. After understanding the basics, the team evaluates current encryption models to see where they fall short—looking at common techniques like RSA, Paillier, and BGV, and also digging into specific security issues like how encrypted data grows, possible attacks, and risks during re-encryption.

II. Fundamentals of Homomorphic Encryption

Homomorphic encryption represents a specialized cryptographic approach that allows direct operations on encrypted data without the need to decrypt it (Alloghani et al., 2019). In this paradigm, computations are executed on ciphertexts, and the corresponding decryption of the result yields the same outcome as if the operation had been performed on the original plaintext. The most valuable variant, fully homomorphic encryption, supports both additive and multiplicative operations enabling a wide range of computations while preserving data confidentiality (Chatterjee & Aung, 2019).

2.1 Definition and Principles

Principles: Homomorphic encryption is fundamentally based on the ability to perform addition and exponential operations directly within the encrypted space (Gouert, Mouris & Tsoutsos, 2023; Gilbert & Gilbert, 2025d). The underlying concept is that any arithmetic operation executed on the ciphertext should, upon decryption, mirror the same operation as if it were performed on the plaintext. This property is maintained through the use of two distinct keys one for encryption and another for decryption a concept reminiscent of the public key infrastructure introduced by RSA (Christopher, 2013; Al Badawi et al., 2022). Thus, the security of homomorphic encryption is closely tied to the robustness of the key management system, drawing parallels to the security mechanisms of RSA (Opoku-Mensah, Abilimi & Amoako, 2013).

Definition: The Paillier public key encryption algorithm is recognized as the first scheme to support additive homomorphism (Zhu et al., 2022). It enables the secure addition of plaintext values within the encrypted domain by utilizing a shared modulus, nnn, to facilitate this operation (Zhu et al., 2022). Although the Paillier scheme allows the summation of encrypted values, it inherently restricts the range of addition based on the predetermined modulus. This limitation ensures that while simple additive operations can be reliably executed, the scheme does not extend to more complex arithmetic operations such as subtraction in the plaintext space without further modifications (Lin et al., 2023).

In summary, homomorphic encryption offers a promising approach to secure cloud computing by enabling encrypted data processing while preserving the confidentiality of sensitive information. Its ability to support secure, multiparty computations and direct operations on ciphertexts positions it as a pivotal technology in the development of privacy-preserving data processing solutions.

2.2. Types of Homomorphic Encryption

Homomorphic encryption (HE) algorithms have come a long way and are now capable of supporting advanced tasks like data mining and machine learning (Wood, Najarian & Kahrobaei, 2020; Gilbert & Gilbert, 2024h). For example, they can assist in performing principal component analysis, especially when key generation is used as a preprocessing step even though this is not considered part of the homomorphic evaluation itself. Past research has also explored a variety of applications, and our own work on threshold-based, somewhat homomorphic encryption for graph-based problems has shown competitive performance and clear efficiency (Chen et al., 2021).

Some challenges in somewhat homomorphic encryption (SHE) have been explored using security game models, including issues such as ciphertext growth, limitations in evaluation capability, and overall design complexity (Hamza et al., 2022). A concept known as the semantic partitioning security assumption has also been introduced to address some of these concerns (Hamza et al., 2022).

Broadly speaking, homomorphic encryption schemes fall into two categories based on the types of computations they can handle (Marcolla et al., Aaraj, 2022). Somewhat homomorphic encryption supports circuits with limited depth essentially low-degree polynomial functions, while fully homomorphic encryption (FHE) can evaluate circuits of any size, including those with arbitrary polynomial depth (Al-Janabi, Al-Janabi & Al-Khateeb, 2023).

SHE can still perform useful functions like parity checks or subset sum evaluations, and these can be applied repeatedly (Gupta & Lakhwani, 2022). However, in practice, the actual capabilities of HE algorithms vary: some are better suited to handling simple computations, while others may struggle, even if theoretical thresholds suggest otherwise.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

III. Key Cryptographic Algorithms

Formally, an HE is a public key encryption scheme with four algorithms: Key Generation – Gen(1k) outputs a pair of a public key pk along with a corresponding secret key sk indexed by a security parameter k (Ragavan & Prabu, 2022). The public key is distributed to the public and the secret key is kept secret. Encryption – Enc(pk,m) with the public key pk and a message m produces a ciphertext c. Decryption – Dec(sk,c) with the private key sk and a ciphertext c returns a message m or a special symbol ⊥ which means the ciphertext does not contain a valid encryption of a message. Circuit Evaluation – Eval(f, c1, c2, . . . , cn) takes an n-ary function f and a number of ciphertexts c1, c2, . . . , cn encrypting the corresponding input m1, m2, . . . , mn for the function f, and outputs a ciphertext that encrypts the result of the f (m1, m2, . . . , mn)(or a special symbol ⊥ if it is not possible to compute the f (m1, m2, . . . , mn). If every circuit representation of any function can be computed under encryption, then we say the fully homomorphic property holds (Ragavan & Prabu, 2019).

Homomorphic encryption (HE) is a cryptographic primitive that enables computation of ciphertexts to an arbitrary function f while only having an encryption of the input x (Gilbert & Gilbert, 2024a). Originally proposed by Rivest, modern HEs are now able to perform efficient arithmetic computations over the encrypted data (Gilbert & Gilbert, 2024g). Similar to public-key encryption (PKE), homomorphic encryption also supports key encapsulation where the public key can be used to get an encapsulated key (similar to encryption) and the corresponding secret key can be used to get the original key (similar to decryption) (Nazeer et al., 2018; Gilbert & Gilbert, 2024y). The difference between PKE and HE is the ability to perform computations on the encrypted data. As a result, PKE and key exchange (KE) schemes have been a building block of a variety of cryptographic applications (Gilbert & Gilbert, 2024x). Similarly, a robust and efficient HE will be a cornerstone of many secure data computation applications.

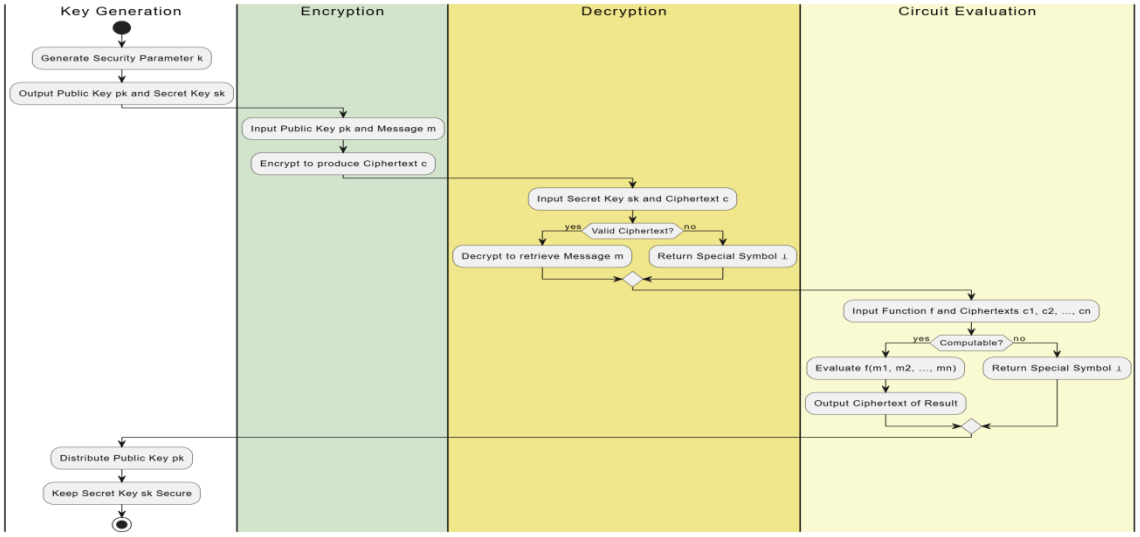

Figure 2: How a homomorphic encryption system works

This diagram shows how homomorphic encryption lets you encrypt data, perform computations on it while it's still encrypted, and only decrypt at the end. That means you can process sensitive data securely without exposing it during computation—perfect for things like cloud computing or privacy-preserving data analysis.

3.1. RSA Encryption

The encryption process is as follows: The transmitter retrieves the public key (n, e). The transmitter converts the message (plaintext) to an integer (M) such that $0 \leq M \leq (n - 1)$ (Eseyin, 2022). The purpose is to map the input space to an acceptable message space. The transmitter computes c, which is the ciphertext, using cryptosystem functions such that $c = M^e \bmod n$. This encryption function is a one-way function, and the receiver uses another system parameter to obtain the plaintext from the encryption. The security comes from the difficulty of resolving the factorization of a large number, which takes an impractical time. It is a fundamental problem of cryptography to find problems that are easy to solve in one direction, but are difficult to solve in the reverse direction (Galla, Koganti & Nuthalapati, 2016; Gilbert & Gilbert, 2024v). RSA is reliable based on this difficulty. In the next subsection, the RSA decryption process is described.

There are two processes, the generation of public key and the encryption. The public key generation process is as follows (Kuppuswamy et al., 2023): Step 1 - Set two prime numbers, p and q, which should be large enough to prevent brute force attacks. Let $n = p \times q$ and $\Phi(n) = (p - 1)(q - 1)$. Step 2 - Choose an integer e such that $1 < e < \Phi(n)$ and e is co-prime to $\Phi(n)$. Step 3 - The decryption key, d, is the multiplicative inverse of e (mod $\Phi(n)$). That is, $d \times e = 1 \bmod \Phi(n)$. The public key is (n, e), and the private key is (n, d). The receiver selects p and q, and creates encryption and decryption keys (Obaid, 2020; Gilbert & Gilbert, 2024w). The private key maximum security of RSA is the factoring of modulus N in a reasonable time. RSA encryption is one of the most widely used methods for securing digital communication (Abilimi et al., 2013). It's an

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

example of what's called an asymmetric encryption system, meaning one key is used to encrypt the data, and a different (secret) key is used to decrypt it (Imam et al., 2021; Abilimi & Adu-Manu, 2013). The security of RSA relies on the mathematical challenge of factoring large numbers. Specifically, the decryption key is derived from the product of two large prime numbers, commonly labeled *p* and *q*.

RSA was introduced in 1978 by three researchers from MIT-Ron Rivest, Adi Shamir, and Leonard Adleman, and the name "RSA" comes from the initials of their last names (Christopher, 2013). At the heart of RSA are modular exponentiation operations, which are key to both the encryption and decryption processes.

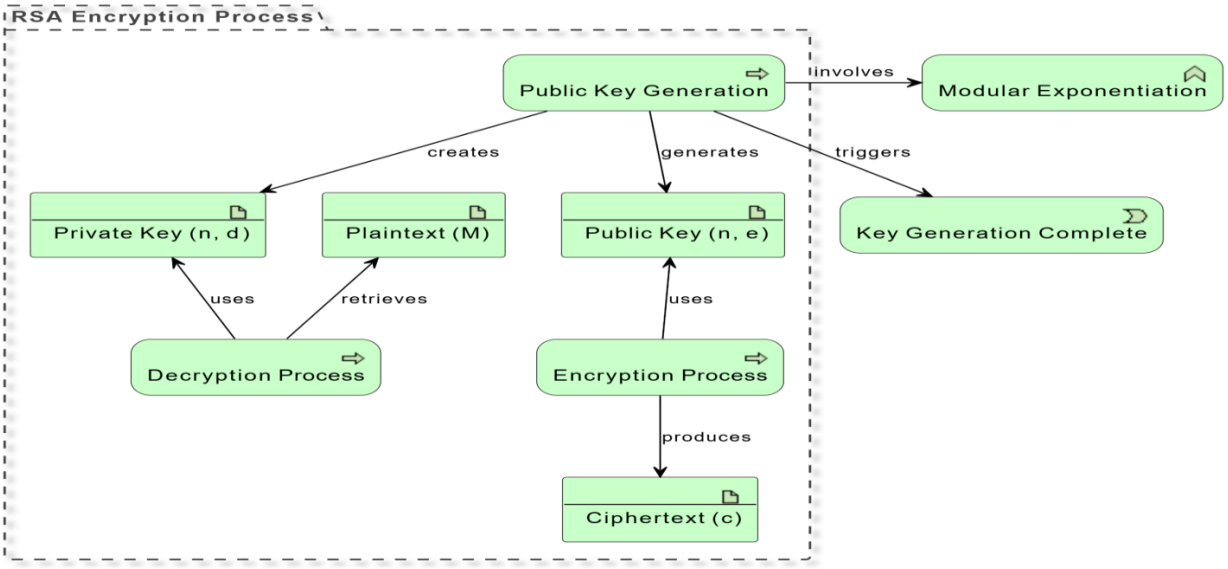

Figure 3: overview of how the RSA encryption process works

This diagram explains how RSA encryption works in a straightforward way. It all starts with creating a public key and a private key kind of like a padlock and a key. The public key is shared with anyone who wants to send you a message, while the private key is kept secret so only you can unlock and read the messages.

3.2. Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is known for providing strong security with relatively short key lengths (Gyamfi, Ansere & Xu, 2019). One of the most common ECC schemes is the Elliptic Curve Public Key Cryptosystem (ECPKC), which relies on the difficulty of solving the elliptic curve discrete logarithm problem (Adeniyi, Jimoh & Awotunde, 2024). Although ECPKC functions as a one-way cryptographic primitive, it still needs to incorporate either randomness or homomorphic properties to be practical in more complex scenarios (Abilimi et al., 2015; Jose & Vijyalakshmi, 2018).

Previous studies that applied basic ECC methods for cloud computing mainly focused on precise computations (Abdaoui et al., 2021). However, these approaches often became more complex even if more effective than simpler ECC-based systems. In this section, we explore ECPKC in more detail, especially how it can be enhanced with homomorphic encryption to support secure and efficient computation.

We also discuss homomorphic encryption as a tool to ensure that cloud users cannot misuse data access. In particular, we look at how homomorphic encryption can be used for various types of data, including applications involving Ethereum. One of our related research projects, in collaboration with the University of Tsukuba, focuses on accelerating the Paillier cryptosystem using GPUs (Yan, 2022). This project, titled "A software package for an asynchronous *fully homomorphic encryption scheme on the GPU for accelerating scientific data analysis"*, is part of a larger cross-ministerial initiative.

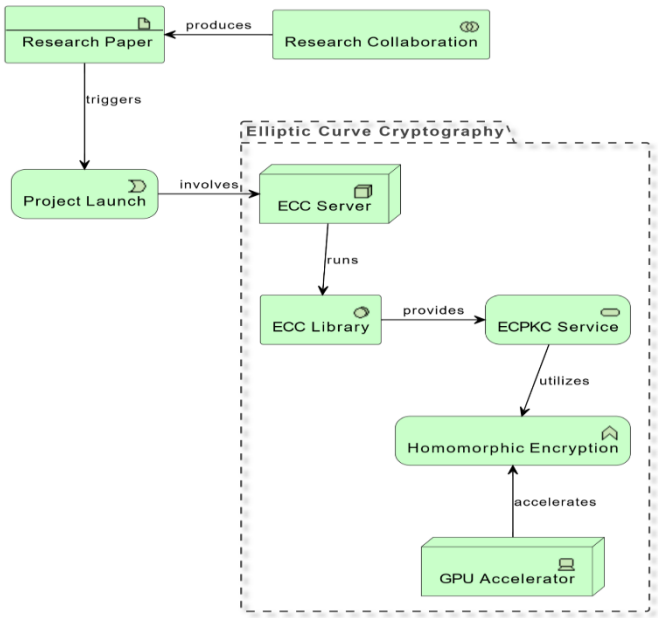

Figure 4: The workflow and technological components on Elliptic Curve Cryptography (ECC) and its application in Homomorphic Encryption.

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

This diagram shows how a research project evolves and the tech behind it. It all starts with a research paper, which comes out of collaboration between researchers. That paper kicks off a new project. As the project gets going, it brings in an ECC (Elliptic Curve Cryptography) server that runs a special ECC library. This library powers a service called ECPKC (basically, a public key crypto service using ECC). That service is then used to enable homomorphic encryption, an advanced way to do computations on encrypted data without needing to decrypt it first.

IV. Secure Data Computation

In this paper, we introduce a new homomorphic encryption approach using lattices with small norm gaps specifically within a Z-scale family of lattices. We also explore how this method performs with Ring-O-based encryption schemes (Gilbert & Gilbert, 2024e). Compared to traditional lattice-based systems, rings provide a better structural fit, making them more efficient for certain algorithms. Thanks to this advantage, our encryption parameters are almost 3.8 times smaller than those in existing n-ring algorithms by Seid (2024). and our approach runs about 100 times faster on average.

These improvements form a solid foundation for more practical and secure cloud computing systems based on lattice cryptography, particularly in fields like computational intelligence (Kaleem et al., 2024).

Traditional cloud computing methods pose risks to data privacy and often provide only limited protection (Sasikumar & Nagarajan, 2024; Gilbert & Gilbert, 2024d). When users upload sensitive data to the cloud, they deserve strong guarantees that their information remains secure even during processing. Some encryption methods allow computations to be performed directly on encrypted data, ensuring privacy throughout the entire process (Seth et al., 2022; Abilimi & Yeboah, 2013). Homomorphic encryption is central to enabling this kind of secure cloud processing.

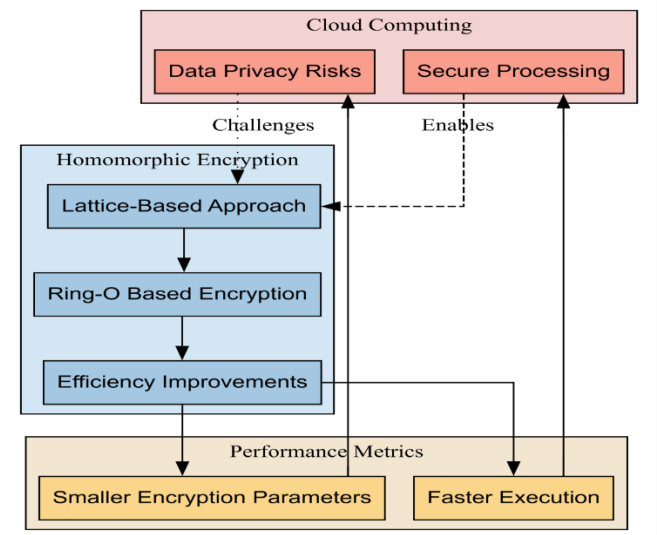

Figure 5: How homomorphic encryption tackles cloud computing challenges and improves performance.

We will plunge deeper into how homomorphic encryption works with standard lattice structures. While earlier approaches showed promise, their practical use was limited by complexity and inefficiency (Pinthurat, Surinkaew & Hredzak, 2024). Our goal is to develop more streamlined, effective encryption techniques that enable secure, verifiable computations without sacrificing performance.

This diagram shows how homomorphic encryption helps solve some of the big challenges in cloud computing—mainly protecting data privacy while still allowing secure processing. It starts with a lattice-based approach, then builds on that with Ring-O encryption, and keeps getting better with ongoing efficiency improvements. These improvements lead to two major benefits: smaller encryption parameters (so it's less heavy) and faster execution (so it runs quicker).

4.1 Challenges and Solutions

This study begins by introducing the fundamentals of homomorphic encryption and explaining the Brakerski-Gentry-Vaikuntanathan (BGV) algorithm. It then explores the main technical challenges and practical implementation strategies associated with this encryption method (Thakur et al., 2025). To address these challenges, the study proposes a fault-tolerant approach based on an enhanced version of the BGV algorithm. By incorporating deep learning, machine learning, and error control coding, the approach demonstrates strong potential for protecting data privacy while also supporting efficient cloud computing (Kamers, 2023; Gilbert & Gilbert, 2024r). The findings suggest that both data security and fault tolerance can be achieved effectively.

One of the core challenges in this field is preserving the original computational behavior of encrypted data without needing to decrypt it essentially treating the encryption system as a "black box" (Albrecht et al., 2021). This idea dates back to the 1970s when Rivest and his colleagues introduced an encryption-based system that allowed two parties to exchange encrypted messages and still compute a shared result. Despite decades of research since then, homomorphic encryption remains largely impractical for broad use, as indicated by Patel (2023). At present, it can handle only basic mathematical operations, and both hardware and software demands remain a significant barrier to widespread adoption.

Unlike traditional encryption, which requires decryption before data can be processed meaningfully, homomorphic encryption allows computations to be performed directly on encrypted data producing results that, once decrypted, are still valid and meaningful (Nita & Mihailescu, 2023). However, current implementations face limitations related to the size of data and the complexity of computations they can support (Kumar & Pabitha, 2024).. These limitations result in significant performance overhead. In the context of cloud computing where analyzing large datasets is common encrypting data, sending it for processing, and decrypting it afterward can be inefficient and risky in terms of privacy (Gilbert & Gilbert, 2024q; Suma & Madhumathy, 2022). Homomorphic encryption aims to solve this, but there's still a long way to go before it becomes fully practical for large-scale, real-world applications.

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

TABLE 1: Challenges and Solutions in Homomorphic Encryption

Challenge	Description	Proposed/Existing Solution
Preservation of computational behavior	Difficulty in computing over encrypted data without decryption (black box computation)	Enhanced BGV algorithm with fault tolerance
Practical implementation of BGV algorithm	Complexity in implementing BGV for real-world use	Use of deep learning, machine learning, and error control coding
Limited computational capabilities	Current schemes support only basic operations; complex computations remain inefficient	Ongoing research and optimization
Performance overhead	Significant slowdown due to processing encrypted data	Improved algorithmic efficiency and system-level optimization
Data size limitations	Large encrypted datasets are difficult to process efficiently	Compression techniques and scalable computation models
Cloud computing inefficiencies	Encrypting, transmitting, and decrypting large datasets is resource-intensive and introduces privacy risks	Homomorphic encryption for secure cloud computation without decryption

Homomorphic encryption is super promising for privacy and security, especially in the cloud; but it's still a work in progress. Researchers are improving algorithms and using modern tech to make it faster, more practical, and more efficient.

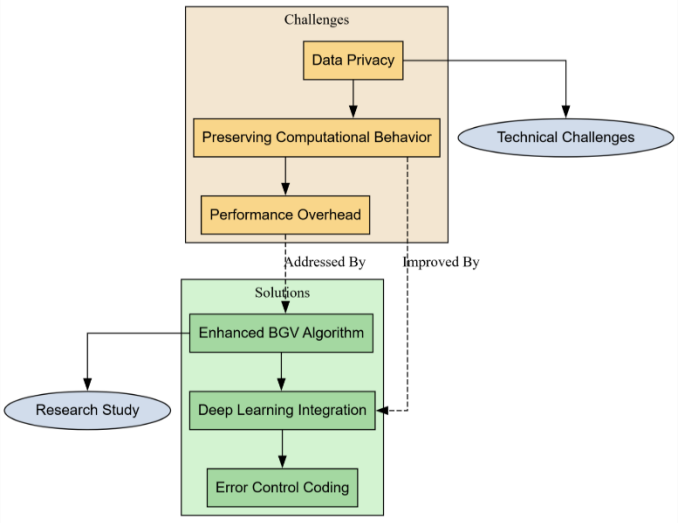

Figure 6: The key technical challenges in secure data processing

Homomorphic encryption is a powerful tool for protecting data privacy, especially in cloud computing, because it lets you perform computations on encrypted data without ever needing to decrypt it. But making it practical and efficient in the real world isn't easy. There are several big challenges: it's hard to compute over encrypted data, implementing complex algorithms like BGV in real systems is tricky, and performance can be slow. Encrypted data is also large and resource-heavy to work with especially in cloud environments where data has to be encrypted, transmitted, and processed securely.

V. Efficiency in Encryption Algorithms

One of the main limitations of homomorphic encryption (HE) is its efficiency. The size of the encrypted output is mainly influenced by two factors: the multiplicative depth and the size of the modulus (Xie et al., 2024). Using large prime numbers can help reduce the modulus size, while multiplicative depth can be optimized through rotation techniques (Singh et al., 2024).

The paper presents an optimized homomorphic algorithm that improves computational efficiency. For example, the multiplicative depth of the intelligent exponentiation operation is 3T, and the size of the long constant involved is 881 bits. However, the multiplication modulus size is 16,360 bits, which is quite large and problematic for practical homomorphic encryption (Munjal & Bhatia, 2023).

In newer schemes like Gen10, the multiplicative depth complexity is less than log(n) + t, making it more efficient. BitDecom-based methods also allow for a homomorphic multiplication depth up to 4t (Alaya, Laouamer & Msilini, 2020). These improvements use smaller multipliers and focus on optimizing lattice-based calculations, particularly by minimizing the size of the least significant bits in the s-parameters (Acar et al., 2018). This significantly reduces the size of the multiplication modulus, leading to faster data processing and more efficient homomorphic operations while still maintaining high expansion rates.

Paillier first introduced a homomorphic encryption scheme based on factoring large composite numbers, which is semantically secure (Alloghani et al., 2019). Later, Gentry proposed a somewhat homomorphic encryption scheme over integers using ideal lattices, capable of performing arbitrary computations on encrypted data (Acar, Uluagac & Conti, 2018). The security of Gentry's approach relies on the difficulty of solving lattice-based problems, particularly finding the shortest vector in a lattice.

Ideal lattices enabled the development of fully homomorphic encryption (FHE), though early versions, like Gentry's original scheme (Acar et al., 2018), required frequent refreshing and re-encryption steps. These schemes support both addition and multiplication on ciphertexts but are only approximately efficient and involve large keys due to inherent limitations in their design (Munjal & Bhatia, 2023; Opoku-Mensah, Abilimi & Boateng, 2013). As a result, they are not yet practical for many real-world applications. Subsequent advancements have proposed more efficient variants, such as Smart's fully homomorphic encryption schemes based on general factorization techniques.

This diagram outlines the development of Homomorphic Encryption (HE), focusing on the challenges of efficiency and key advancements. It shows how issues like modulus size and multiplicative depth led to innovations such as optimized algorithms, rotation techniques, and Gentry's groundbreaking scheme, which enabled Fully Homomorphic Encryption (FHE). While FHE is secured by hard lattice problems, it still faces

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

design limitations that have driven ongoing improvements in the field.

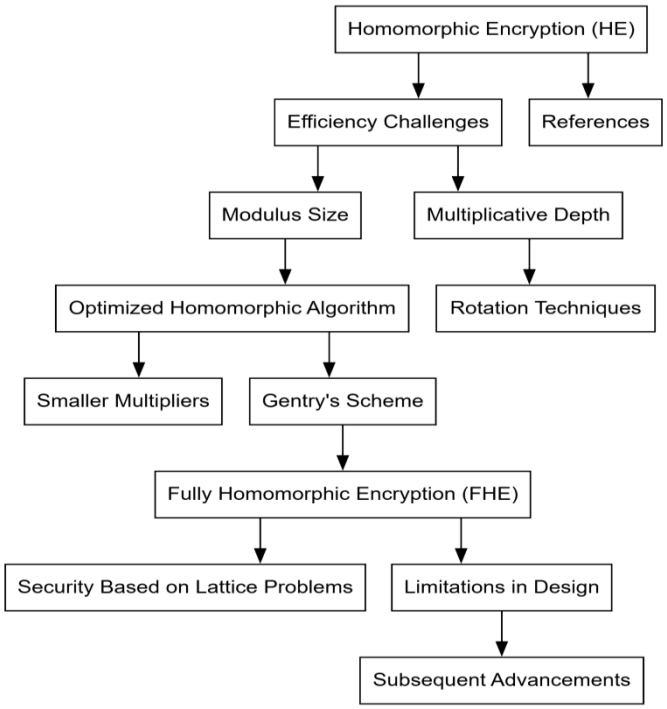

Figure 7: Illustrating the efficiency in Encryption Algorithms

5.1 Optimization Techniques

In this paper, we also explore various algorithmic optimization techniques that enhance the performance of homomorphic encryption (HE) schemes. We start by looking at how the encryption process can be made more efficient by leveraging the algebraic properties of modular addition and integer multiplication (Doan et al., 2023). We then examine how Fast Fourier Transform (FFT) methods help accelerate the relinearization phase of computation. Lastly, we review strategies for optimizing modular reductions and briefly touch on techniques used during the downsizing stage of the scheme (Jeniffer & Chandrasekar, 2022).

Over the years, the performance of HE schemes has significantly improved thanks to extensive research and development. These improvements often focus on making decryption faster and maximizing the utility of ciphertexts (Xie et al., 2024). This is achieved through a mix of mathematical, algorithmic, and hardware optimizations, all tailored to the specific features of the target platform. Key techniques include refining existing cryptographic algorithms and using more efficient algebraic operations, for example, tapping into useful properties of ring structures (Jeniffer & Chandrasekar, 2022). Another major focus is managing noise, which accumulates during encrypted computations. By controlling noise levels at each modular step, it becomes easier to manage evaluation keys, resulting in faster processing and lower memory usage during encryption (Doan, Messai et al., 2023).

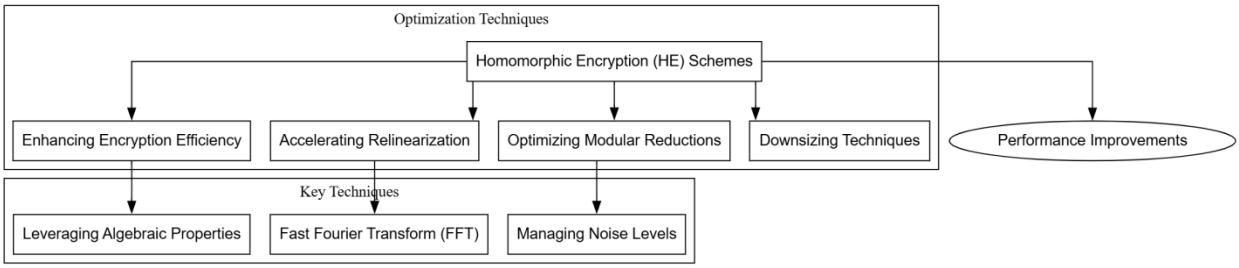

Figure 8: Various Optimization Techniques.

This diagram highlights various optimization techniques used to improve the performance of Homomorphic Encryption (HE) schemes. It focuses on enhancing encryption efficiency, accelerating relinearization, optimizing modular reductions, and applying downsizing techniques. Key methods supporting these improvements include leveraging algebraic properties, using Fast Fourier Transform (FFT), and managing noise levels. Altogether, these strategies aim to make HE more practical and efficient for real-world applications.

VI. Homomorphic Encryption in Cybersecurity

All cryptographic methods have certain limitations, often due to the nature of the algorithms or the structure of the data they protect (Xie et al., 2024; Gilbert et al., 2025). In areas like biometric data protection or input validation, many traditional algorithms fall short (Yeboah, Opoku-Mensah & Abilimi, 2013b). This is where property-preserving encryption becomes valuable it ensures that certain essential characteristics of the data remain intact even during encrypted processing, which is crucial for applications like Big Data analysis and secure computations over encrypted data (Sawant, 2022; Gilbert & Gilbert, 2024u; Jeniffer & Chandrasekar, 2022).

Classical cryptanalysis is notoriously difficult, and homomorphic encryption helps address one of the key challenges in cybersecurity (Gilbert & Gilbert, 2924t; Amorim & Costa, 2023); preventing unauthorized access and misuse of sensitive data, especially in environments where multiple parties are involved (Doan et al., 2023; Gilbert & Gilbert, 2024b). By using HE, we can allow computations on encrypted data without revealing the data itself making it ideal for distributed and cloud-based systems. In these settings, data remains encrypted while stored in the cloud, and users can run encrypted queries that preserve the original properties of the data (Chavarín et al., 2023; Shankar & Lakshmanaprabu, 2018; Gilbert & Gilbert, 2024n). These property-preserving searches are performed directly on the encrypted information, and repeated queries can yield the results users need without compromising privacy or security (Gilbert & Gilbert, 2024c).

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

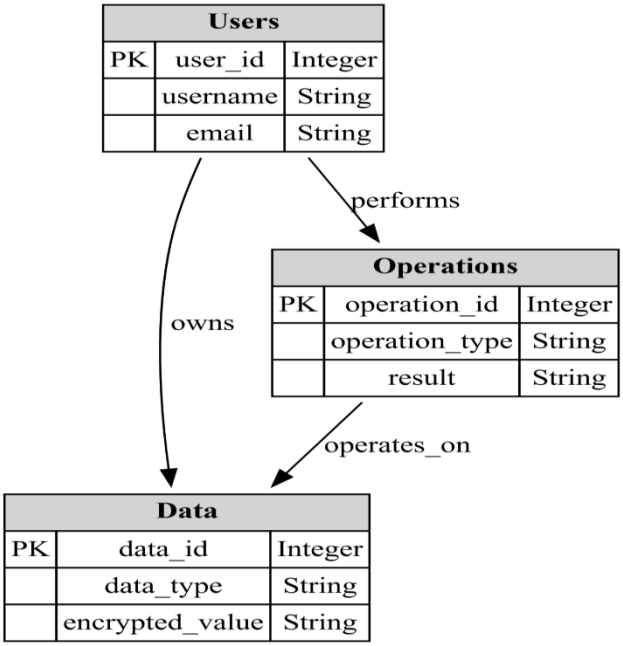

Figure 8: Representation of Homomorphic Encryption in Cybersecurity

This diagram represents a simple database schema that models interactions between users, operations, and encrypted data. Each user, identified by a unique user ID, can own multiple pieces of encrypted data and perform various operations. Operations, identified by an operation ID, include a type and a result, and they act upon specific data entries. The data itself is characterized by a data ID, its type, and an encrypted value. The relationships show how users perform operations and own data, while operations are carried out on specific data items.

6.1. Applications and Use Cases

Encrypted medical data comes with its own set of challenges like privacy concerns, data residency rules, and restrictions on how data is moved or accessed. But thanks to encryption technologies, many of these limitations can now be addressed (Jin et al., 2019). As a result, users feel more confident using cloud services, and organizations have a clearer responsibility to keep sensitive data secure. Encryption also makes it possible to outsource tasks like BioSQL encryption, letting bioinformatics tools be broken down into modules that can be managed and updated more easily (Mahapatra, Krishnamurthi & Nayyar, 2019).

A particularly interesting use case is in cloud-based healthcare services. In this setup, users don't need to fully trust the cloud provider but can still run secure medical algorithms remotely. Here's how it works: a patient uploads encrypted DNA data along with a function to be executed using a homomorphic encryption (HME) algorithm (Nowrozy, 2024). The cloud provider, assumed to be honest but curious, processes the encrypted data using the algorithm—without ever seeing the data in its original form. The encrypted results are then returned to the user, who can decrypt them locally. This way, only the data owner can read the final outcome (Nowrozy, 2024).

Different types of homomorphic encryption come with trade-offs. For instance, additive schemes like ElGamal require key lengths that grow quickly, making them slower when updated or re-keyed (Jin et al., 2019). On the other hand, multiplicative schemes such as BGV need to be large enough to handle the range of possible outputs without issues like "wrapping," which can make the encryption unreliable. That's because each multiplication causes a 10–12x increase in data size. Somewhat homomorphic encryption schemes, which are limited in the number of operations they can perform, are more efficient for smaller or simpler tasks and don't need complex workarounds (Jin et al., 2019).

Before choosing an encryption method, it's crucial to look at the specific needs of your use case. The right solution depends on what you're trying to protect and how the data will be used (Yigzaw et al., 2022). The next section walks through examples of encryption and decryption in action, followed by a deeper look at the technical requirements for homomorphic encryption and other options.

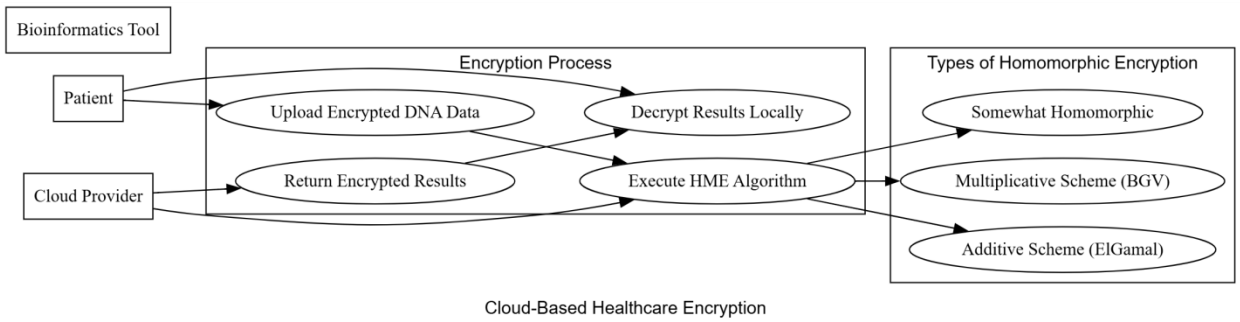

Figure 9: Cloud-Based Healthcare Encryption.

This diagram illustrates a cloud-based healthcare encryption system using homomorphic encryption. A patient uses a bioinformatics tool to upload encrypted DNA data to a cloud provider. The cloud provider then executes a homomorphic encryption (HME) algorithm to process the data without decrypting it. Once processed, encrypted results are returned to the patient, who decrypts them locally. The system supports different types of homomorphic encryption—somewhat homomorphic, multiplicative (BGV), and additive (ElGamal)—to enable secure and private genomic analysis in the cloud.

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

VII. Case Studies and Implementations

To better understand how these algorithms perform in practice, we implemented several homomorphic encryption and decryption schemes and measured their performance. Libraries like HEAAN, which use techniques such as full-barreled modulo rings with six primes and kPaillier encryption, showed strong results—delivering smaller ciphertexts and faster processing times (Al Badawi et al., 2019).

Another standout is the HElib library, which uses the Nussbaumer convolution algorithm for efficient polynomial arithmetic. It converts data from number-theoretic form to a polynomial module format in logarithmic time. Compared to other libraries like Palisade and Microsoft SEAL, HElib performs especially well because it applies modulus switching—a technique that enhances both encoding and core operations like addition, multiplication, and convolution (Alloghani et al., 2019).

Overall, a library's efficiency is influenced not just by encryption and decryption speed, but also by how well it supports complex features, the ease of use of its tools, and its general user-friendliness. In this section, we also critically evaluate the state-of-the-art in secure computation algorithms, including those that support many-to-one and one-to-many data relationships.

7.1 Real-world Examples

Homomorphic encryption was first introduced by Rivest, Adleman, and Dertouzos, and it has since become a transformative approach to data security (Chatterjee & Aung, 2019). It enables computations to be performed on encrypted data without needing to decrypt it first. This means sensitive data remains secure throughout the process, and the server conducting the operations never sees the underlying data (Mollakuqe et al., 2024).

Why is this important? As data volumes grow, it's no longer practical or safe for humans to handle sensitive computations manually. Servers offer the speed and scalability required to process large datasets, but only if those computations can be done securely (Tekin, 2023; Gilbert, Oluwatosin & Gilbert, 2024). Homomorphic encryption makes this possible. It allows users to outsource complex computations to remote servers without sacrificing privacy or introducing security risks (Al-Janabi, Al-Janabi & Al-Khateeb, 2023).

For businesses and researchers, this means they can work with massive amounts of encrypted data, run analytics, and generate insights all without exposing any private information. It's a win-win: improved performance without compromising security (Chatterjee & Aung, 2019).

VIII. Research Directions

As technology evolves, so do the needs for protecting information. Right now, homomorphic encryption (HE) shows a lot of promise for data security, but it still faces big limitations especially when it comes to handling complex computations efficiently (Chahar, 2025). Moving forward, researchers, industry experts, and policymakers need to collaborate to build practical, secure, and scalable HE algorithms that can be used in real-world scenarios.

The challenge lies in turning today's tough, resource-heavy problems into smaller, manageable tasks. Future efforts shouldn't just focus on improving existing HE algorithms (Mohamed, 2025). We also need to think about how regular people interact with encrypted systems like how easy they are to use, whether users understand what's happening, and how they behave when using encrypted smartphones or big data services (Zafir et al., 2024). That's where ideas like "living labs" and participatory research come in they help bridge the gap between technical innovation and everyday use.

As privacy-preserving computation becomes more important, we'll need systems that can securely and efficiently handle both encrypted and unencrypted data. This includes verifying data integrity in both forms (Taherdoost, Le & Slimani, 2025). In many cases, data needs to stay encrypted even while being processed, which is a big technical hurdle. But solving this will open the door for secure, real-time data interactions and sharing, especially in environments where multiple parties are working together (Taherdoost, Le & Slimani, 2025).

One key area to explore is making HE work better with big data frameworks like Map Reduce. Big data environments demand flexibility, speed, and resilience from encryption algorithms. (Taherdoost, Le & Slimani, 2025). That means designing custom HE schemes tailored to specific use cases, and blending cryptography with other tech like machine learning, data compression, and algorithm optimization (Gilbert & Gilbert, 2024s). Also, we should work on expanding the capabilities of partially homomorphic encryption and enhancing cryptosystems that support more advanced mathematical functions like Fourier transforms (Gilbert & Gilbert, 2024m; Marcolla et al., 2022).

The biggest challenge is figuring out how to design HE algorithms that can handle these demanding tasks without sacrificing performance. There are several promising directions for future research.

8.1. Emerging Technologies

One advantage of HE is that it keeps the encryption details hidden from the cloud provider, the data center, and even the user. It's invisible by design. Recent cryptographic advances have improved HE in terms of privacy like ensuring the computations are secure even if the service provider is only "semi-honest" (that is; they follow the rules but might try to learn something from the data) (Sehgal & Bhatt, 2018; Gilbert, 2021).

We're now seeing HE used in things like machine learning applications, where the models can make predictions on encrypted data without needing to decrypt it first (Gilbert, 2012). This is showing up in academic studies, security tools, and plans from top cloud providers. It's a fast-moving area, so we need to keep a close eye on how it develops and stay alert to the new risks it might bring (Sehgal, Bhatt & Acken, 2020; Gilbert, 2022).

The original vision of HE, dating back to the 1970s, was to allow computations on encrypted data, without needing to decrypt it first. That was revolutionary. It opened doors for things like secure flight control, encrypted data backups, digital

forensics, and even secure e-voting (Gilbert & Gilbert, 2024l). Today, HE is becoming more mature. We're finally seeing efficient software implementations that make it usable in real systems. Early on, HE was mostly about running automated tasks where the delay in decryption didn't matter too much. But as the technology improves, it's starting to look like a practical tool for a wider range of real-time, privacy-sensitive applications (Tyagi, 2023).

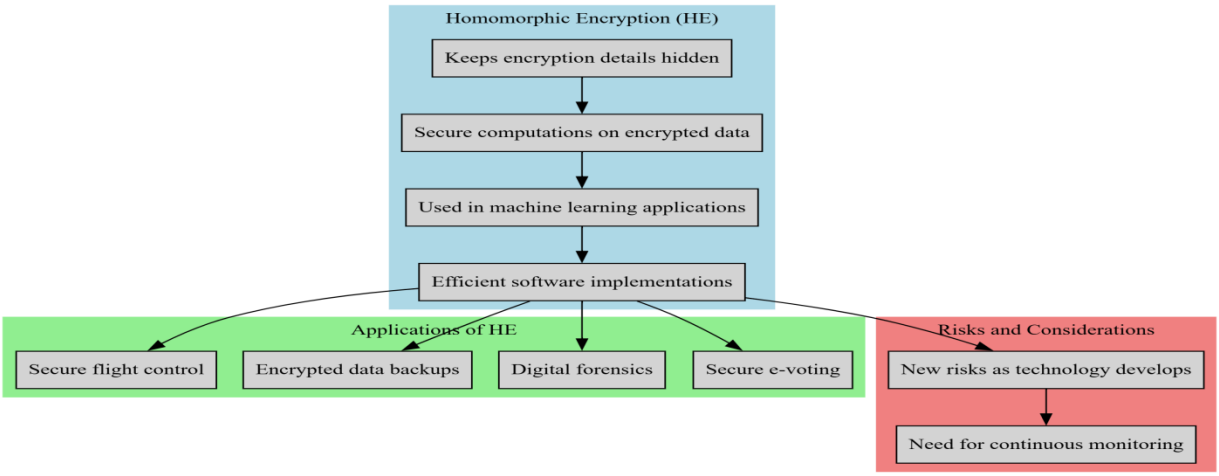

Figure 10: Homomorphic Encryption Emerging Technologies

This diagram explains the role of Homomorphic Encryption (HE) by highlighting its core benefits, real-world applications, and associated risks. HE allows secure computations on encrypted data without revealing sensitive information, making it valuable for machine learning and other software implementations. Its applications span secure flight control, encrypted data backups, digital forensics, and secure e-voting. However, as the technology evolves, it brings new risks that demand continuous monitoring and evaluation to ensure safety and reliability.

IX. Findings, Conclusions, Recommendations and Future Trends

This study revealed that homomorphic encryption (HE) holds tremendous potential for enhancing data security in cloud-based environments. Through real-world implementation and evaluation of multiple HE schemes including Paillier, BGV, and lattice-based systems the research demonstrated that it is indeed possible to perform secure computations directly on encrypted data without exposing sensitive information.
Key findings include:

- Functional Security: HE schemes like BGV and Paillier effectively support secure arithmetic operations on ciphertext, with fully homomorphic encryption (FHE) enabling both addition and multiplication.
- Efficiency Gaps: While current HE algorithms show strong security guarantees, their practical performance is hindered by high computational overhead, large ciphertext sizes, and limits on the complexity of supported operations.
- Geometric Encryption Innovations: A novel approach integrating geometric structures with HE demonstrated promising efficiency improvements, particularly in multi-party scenarios.
- Optimization Success: Techniques such as Fast Fourier Transforms, modular reduction, and relinearization significantly improved processing times, suggesting that HE can be fine-tuned for better real-world applicability.
- Applications in Sensitive Domains: The case studies showed that HE can be effectively used in domains such as healthcare and finance, where secure processing of private data is critical. Notably, encrypted medical computations performed in the cloud were both secure and functional, even without revealing raw data.

X. Conclusions

Homomorphic encryption stands out as a transformative technology that addresses one of the major drawbacks of traditional encryption: the inability to process data while it remains encrypted. By enabling computation over ciphertext, HE ensures that privacy is maintained throughout the entire data lifecycle from storage to analysis.

However, the study also makes it clear that HE, in its current state, is not without limitations. Most schemes suffer from performance bottlenecks and require careful calibration to balance security with usability. Despite these issues, advancements in optimization techniques and algorithm design are steadily pushing HE toward practical deployment.

Ultimately, this research confirms that while HE is not yet a universal solution, it is a powerful tool in the growing field of privacy-preserving computation. With continued development, it has the potential to reshape the way we think about secure data processing in cloud and distributed systems.

Recommendations

Based on the findings, the study makes the following recommendations:

- Invest in Optimization: Researchers and developers should focus on refining current HE schemes by minimizing

Chris Gilbert and Mercy Abiola Gilbert, "Homomorphic Encryption Algorithms for Secure Data Computation," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 2, pp. 148-162, 2025.

ciphertext size and computational depth. Lightweight implementations are key to widespread adoption.

- Tailor HE for Specific Use Cases: Not all data operations require fully homomorphic functionality. Developing custom HE solutions for domain-specific tasks (e.g., medical diagnostics, financial analytics) can lead to more efficient systems.
- Promote Interdisciplinary Collaboration: Bridging cryptography with fields like machine learning, data science, and cloud infrastructure can accelerate the integration of HE in practical applications.
- Prioritize Usability: For HE to become mainstream, it must be user-friendly. Designing intuitive tools, interfaces, and documentation is essential, especially for non-expert users in sensitive industries.
- Strengthen Fault Tolerance: Incorporating error-correcting codes and resilience strategies will help HE schemes perform more reliably in unpredictable real-world computing environments.

Future Trends and Research Directions

Looking ahead, the future of homomorphic encryption is promising, yet challenging. Several key areas are poised for further exploration:

- Scalability in Big Data Frameworks: Integrating HE with big data platforms like Hadoop or Spark (e.g., via MapReduce) is essential to handle real-time processing of massive encrypted datasets.
- HE in AI and Machine Learning: There is growing interest in privacy-preserving machine learning, where models are trained and deployed on encrypted data—especially useful in healthcare and finance.
- Combining HE with Other Privacy Technologies: Blending HE with zero-knowledge proofs, secure multiparty computation (SMPC), or differential privacy can create more robust and flexible security architectures.
- Improved Support for Complex Operations: Future work should explore how HE can handle advanced mathematical computations, such as matrix operations, Fourier transforms, and gradient descent, without significant performance loss.
- Hardware Acceleration: Leveraging GPU and specialized hardware (example, FPGA or ASIC) will play a critical role in making HE fast enough for production use.
- Policy and Governance: As HE becomes more widespread, there will be a need to establish regulatory frameworks and ethical guidelines for its deployment—particularly where user consent and data sovereignty are involved.

In sum, homomorphic encryption is advancing from theoretical promise to practical application. Continued innovation and collaboration across disciplines will be vital in realizing its full potential for secure, real-time data processing.

REFERENCES

1. Abdaoui, A., Erbad, A., Al-Ali, A. K., Mohamed, A., & Guizani, M. (2021). Fuzzy elliptic curve cryptography for authentication in Internet of Things. *IEEE Internet of Things Journal, 9*(12), 9987–9998.
2. Abilimi,C.A, Asante,M, Opoku-Mensah, E & Boateng, F.O. (2015). Testing for Randomness in Pseudo Random Number Generators Algorithms in a Cryptographic Application.Computer Engineering and Intelligent Systems, www.iiste.org, ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online) Vol.6, No.9, 2015
3. Abilimi, C. A., & Adu-Manu, K. S. (2013). Examining the impact of Information and Communication Technology capacity building in High School education in Ghana. International Journal of Engineering Research & Technology (IJERT),ISSN: 2278-0181,Vol. 2 Issue 9, September - 2013
4. Abilimi, C.A., Amoako, L., Ayembillah, J. N., Yeboah, T.(2013). Assessing the Availability of Information and Communication Technologies in Teaching and Learning in High School Education in Ghana. International Journal of Engineering Research and Technology, 2(11), 50 - 59.
5. Abilimi, C. A. & Yeboah, T. (2013). Assessing the challenges of Information and Communication Technology in educational development in High Schools in Ghana. International Journal of Engineering Research & Technology (IJERT).ISSN: 2278-0181, Vol. 2 Issue 11, November - 2013
6. Adeniyi, A. E., Jimoh, R. G., & Awotunde, J. B. (2024). A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security. *Computers and Electrical Engineering, 118*, 109330.
7. Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D. B., Erabelli, S., Genise, N., ... & Zucca, V. (2022, November). Openfhe: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography* (pp. 53–63).
8. Al Badawi, A., Polyakov, Y., Aung, K. M. M., Veeravalli, B., & Rohloff, K. (2019). Implementation and performance evaluation of RNS variants of the BFV homomorphic encryption scheme. *IEEE Transactions on Emerging Topics in Computing, 9*(2), 941–956.
9. Al-Janabi, A. A., Al-Janabi, S. T. F., & Al-Khateeb, B. (2023). Secure data computation using deep learning and homomorphic encryption: A survey. *International Journal of Online & Biomedical Engineering, 19*(11).
10. Alaya, B., Laouamer, L., & Msilini, N. (2020). Homomorphic encryption systems statement: Trends and challenges. *Computer Science Review, 36*, 100235.
11. Alloghani, M., Alani, M. M., Al-Jumeily, D., Baker, T., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2019). A systematic review on the status and progress of homomorphic encryption technologies. *Journal of Information Security and Applications, 48*, 102362.
12. Amorim, I., & Costa, I. (2023). Leveraging searchable encryption through homomorphic encryption: A comprehensive analysis. *Mathematics, 11*(13), 2948.
13. Bagchi, P., Bera, B., Das, A. K., Shetty, S., Vijayakumar, P., & Karuppiah, M. (2023). Post quantum lattice-based secure framework using aggregate signature for ambient intelligence assisted blockchain-based IoT applications. *IEEE Internet of Things Magazine, 6*(1), 52–58.
14. Bishukarma, R. (2023). Privacy-preserving based encryption techniques for securing data in cloud computing environments. *Int. J. Sci. Res. Arch, 9*(2), 1014–1025.
15. Chatterjee, A., & Aung, K. M. M. (2019). *Fully homomorphic encryption in real world applications*. Singapore: Springer.
16. Chavarín, Á., Cuevas, E., Avalos, O., Gálvez, J., & Pérez-Cisneros, M. (2023). Contrast enhancement in images by homomorphic filtering and cluster-chaotic optimization. *IEEE Access, 11*, 73803–73822.
17. Chen, Z., Hu, G., Zheng, M., Song, X., & Chen, L. (2021). Bibliometrics of machine learning research using homomorphic encryption. *Mathematics, 9*(21), 2792.
18. Christopher, A. A.(2013). Effective Information Security Management in Enterprise Software Application with the Revest-Shamir-Adleman (RSA) Cryptographic Algorithm.International Journal of Engineering Research & Technology (IJERT),ISSN: 2278-0181,Vol. 2 Issue 8, August - 2013.
19. Doan, T. V. T., Messai, M. L., Gavin, G., & Darmont, J. (2023). A survey on implementations of homomorphic encryption schemes. *The Journal of Supercomputing, 79*(13), 15098–15139.
20. Eseyin, J. B. (2022). *Enhanced asymmetric data encryption algorithms using residue number system and steganography* (Doctoral dissertation, Kwara State University [Nigeria]).

21. Galla, L. K., Koganti, V. S., & Nuthalapati, N. (2016, December). Implementation of RSA. In *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)* (pp. 81–87). IEEE.
22. Gilbert, C.(2012). The Quest of Father and Son: Illuminating Character Identity, Motivation, and Conflict in Cormac McCarthy's The Road. English Journal, Volume 102, Issue Characters and Character, p. 40 - 47. https://doi.org/10.58680/ej201220821.
23. Gilbert, C. (2018). Creating Educational Destruction: A Critical Exploration of Central Neoliberal Concepts and Their Transformative Effects on Public Education. The Educational Forum, 83(1), 60–74. https://doi.org/10.1080/00131725.2018.1505017.
24. Gilbert, C. (2021). Walking the popular education spiral - an account and analysis of participatory action research with teacher activists. Educational Action Research, 30(5), 881–901. https://doi.org/10.1080/09650792.2021.1875856
25. Gilbert, C. (2022). Making the Invisible Visible: Professional Development to Support Teacher Activism. Kappa Delta Pi Record, 58(1), 14–19. https://doi.org/10.1080/00228958.2022.2005426
26. Gilbert, C. & Gilbert, M.A.(2024a). Unraveling Blockchain Technology: A Comprehensive Conceptual Review. International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and ISSN Approved), ISSN:2349-5162, Vol.11, Issue 9, page no. ppa575-a584, September-2024, Available at : http://www.jetir.org/papers/JETIR2409066.pdf
27. Gilbert, C. & Gilbert, M.A.(2024b). Strategic Framework for Human-Centric AI Governance: Navigating Ethical, Educational, and Societal Challenges. International Journal of Latest Technology in Engineering Management & Applied Science, 13(8), 132-141. https://doi.org/10.51583/IJLTEMAS.2024.130816
28. Gilbert, C. & Gilbert, M.A.(2024c). The Impact of AI on Cybersecurity Defense Mechanisms: Future Trends and Challenges.Global Scientific Journals.ISSN 2320-9186,12(9),427-441. https://www.globalscientificjournal.com/researchpaper/The_Impact_of_AI_on_Cybersecurity_Defense_Mechanisms_Future_Trends_and_Challenges_.pdf.
29. Gilbert, C. & Gilbert, M.A. (2024d). The Convergence of Artificial Intelligence and Privacy: Navigating Innovation with Ethical Considerations. *International Journal of Scientific Research and Modern Technology*, *3*(9), 9-9.
30. Gilbert, C. & Gilbert, M.A.(2024e). Transforming Blockchain: Innovative Consensus Algorithms for Improved Scalability and Security. International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.11, Issue 10, page no.b299-b313, October-2024, Available :http://www.jetir.org/papers/JETIR2410134.pdf
31. Gilbert, C. & Gilbert, M.A. (2024f). Future Privacy Challenges: Predicting the Agenda of Webmasters Regarding Cookie Management and Its Implications for User Privacy. International Journal of Advanced Engineering Research and Science, ISSN (Online): 2455-9024,Volume 9, Issue 4, pp. 95-106.
32. Gilbert, C., & Gilbert, M. A. (2024g). Navigating the Dual Nature of Deepfakes: Ethical, Legal, and Technological Perspectives on Generative Artificial Intelligence (AI) Technology. *International Journal of Scientific Research and Modern Technology*, *3*(10). https://doi.org/10.38124/ijsrmt.v3i10.54
33. Gilbert, C., & Gilbert, M. A. (2024h). Revolutionizing Computer Science Education: Integrating Blockchain for Enhanced Learning and Future Readiness. International Journal of Latest Technology in Engineering, Management & Applied Science, ISSN 2278-2540, Volume 13, Issue 9, pp.161-173.
34. Gilbert, C. & Gilbert, M.A. (2024i). Unlocking Privacy in Blockchain: Exploring Zero-Knowledge Proofs and Secure Multi-Party Computation Techniques. Global Scientific Journal (ISSN 2320-9186) 12 (10), 1368-1392.
35. Gilbert, C. & Gilbert, M.A. (2024j). The Role of Artificial Intelligence (AI) in Combatting Deepfakes and Digital Misinformation. International Research Journal of Advanced Engineering and Science (ISSN: 2455-9024), Volume 9, Issue 4, pp. 170-181.
36. Gilbert, C. & Gilbert, M.A.(2024k). AI-Driven Threat Detection in the Internet of Things (IoT), Exploring Opportunities and Vulnerabilities. International Journal of Research Publication and Reviews, Vol 5, no 11, pp 219-236.
37. Gilbert, C., & Gilbert, M. A. (2024l). The security implications of artificial intelligence (AI)-powered autonomous weapons: Policy recommendations for international regulation. *International Research Journal of Advanced Engineering and Science*, *9*(4), 205–219.
38. Gilbert, C., & Gilbert, M. A. (2024m). The role of quantum cryptography in enhancing cybersecurity. *International Journal of Research Publication and Reviews*, *5*(11), 889–907. https://www.ijrpr.com
39. Gilbert, C., & Gilbert, M. A. (2024n). Bridging the gap: Evaluating Liberia's cybercrime legislation against international standards. *International Journal of Research and Innovation in Applied Science (IJRIAS)*, *9*(10), 131–137. https://doi.org/10.51584/IJRIAS.2024.910013
40. Gilbert, C., & Gilbert, M. A. (2024o). The Effectiveness of Homomorphic Encryption in Protecting Data Privacy. *International Journal of Research Publication and Reviews*, *5*(11), 3235-3256. https://www.ijrpr.com.
41. Gilbert, C., & Gilbert, M. A. (2024p). Cryptographic Foundations and Cybersecurity Implications of Blockchain Technology. *Global Scientific Journals*,ISSN 2320-9186,12(11),464-487. https://www.globalscientificjournal.com
42. Gilbert, C., & Gilbert, M. A. (2024q). Advancing privacy standards through education: The role of academic initiatives in enhancing privacy within Cardano's blockchain ecosystem. *International Research Journal of Advanced Engineering and Science*, *9*(4), 238–251.
43. Gilbert, C., & Gilbert, M. A. (2024r). Leveraging artificial intelligence (AI) by a strategic defense against deepfakes and digital misinformation. *International Journal of Scientific Research and Modern Technology*, *3*(11). https://doi.org/10.38124/ijsrmt.v3i11.76
44. Gilbert, C., & Gilbert, M. A. (2024s). Evaluation of the efficiency of advanced number generators in cryptographic systems using a comparative approach. *International Journal of Scientific Research and Modern Technology*, *3*(11). https://doi.org/10.38124/ijsrmt.v3i11.77
45. Gilbert, C., & Gilbert, M. A. (2024t). Cybersecurity risk management frameworks for critical infrastructure protection. *International Journal of Research Publication and Reviews*, *5*(12), 507–533. https://www.ijrpr.com/
46. Gilbert, C., & Gilbert, M. A. (2024u). Organizational and leadership aspects of cybersecurity governance. *International Journal of Research Publication and Reviews*, *5*(12), 1174–1191. Retrieved from www.ijrpr.com
47. Gilbert, C., & Gilbert, M. A. (2024v). The development and evolution of cryptographic algorithms in response to cyber threats. *International Journal of Research Publication and Reviews*, *5*(12), 1149–1173. Retrieved from www.ijrpr.com
48. Gilbert, C., & Gilbert, M. A. (2024w). Privacy-preserving data mining and analytics in big data environments. *Global Scientific Journal*, *12*(12). Retrieved from www.globalscientificjournal.com
49. Gilbert, C., & Gilbert, M. A. (2024x). Investigating the challenges and solutions in cybersecurity using quantum computing and cryptography. International Research Journal of Advanced Engineering and Science, 9(4), 291–315.
50. Gilbert, C., & Gilbert, M. A. (2024y). The integration of blockchain technology into database management systems for enhanced security and transparency. International Research Journal of Advanced Engineering and Science, 9(4), 316–334.
51. Gilbert, C., & Gilbert, M. A. (2025a). *Artificial intelligence (AI) and machine learning (ML) for predictive cyber threat intelligence (CTI)*. International Journal of Research Publication and Reviews, 6(3), 584–617. http://www.ijrpr.com
52. Gilbert, C., & Gilbert, M. A. (2025b). Continuous user authentication on mobile devices. International Research Journal of Advanced Engineering and Science, 10(1), 158–173.
53. Gilbert, C., & Gilbert, M. A. (2025c). *Patterns and vulnerabilities of cryptocurrency-related cybercrimes*. *Global Scientific Journal*, 13(3), 1950-1981. https://www.globalscientificjournal.com
54. Gilbert, C., & Gilbert, M. A. (2025d). *Data encryption algorithms and risk management*. *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*, *14*(3), 479–497. https://doi.org/10.51583/IJLTEMAS.2025.140300054
55. Gilbert, C., Gilbert, M. A., Dorgbefu, M., Leakpor, D. J., Gaylah, K. D., & Adetunde, I. A. (2025). Enhancing detection and response using artificial intelligence in cybersecurity. *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, *7*(10), 87-104.

56. Gilbert, M.A., Oluwatosin, S. A., & Gilbert, C. (2024). An investigation into the types of role-based relationships that exist between lecturers and students in universities across southwestern nigeria: a sociocultural and institutional analysis. Global Scientific Journal, ISSN 2320-9186, Volume 12, Issue 10, pp. 263-280.
57. Gilbert, M.A., Auodo, A. & Gilbert, C. (2024). Analyzing Occupational Stress in Academic Personnel through the Framework of Maslow's Hierarchy of Needs. International Journal of Research Publication and Reviews, Vol 5, no 11, pp 620-630.
58. Gouert, C., Mouris, D., & Tsoutsos, N. (2023). SOK: New insights into fully homomorphic encryption libraries via standardized benchmarks. *Proceedings on Privacy Enhancing Technologies*.
59. Gupta, G., & Lakhwani, K. (2022). An enhanced approach to improve the encryption of big data using intelligent classification technique. *Multimedia Tools and Applications, 81*(18), 25171–25204.
60. Gupta, I., Singh, A. K., Lee, C. N., & Buyya, R. (2022). Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions. *IEEE Access, 10*, 71247–71277.
61. Hamza, R., Hassan, A., Ali, A., Bashir, M. B., Alqhtani, S. M., Tawfeeg, T. M., & Yousif, A. (2022). Towards secure big data analysis via fully homomorphic encryption algorithms. *Entropy, 24*(4), 519.
62. Hussain, S., Farooq, M., Alzahrani, B. A., Albeshri, A., Alsubhi, K., & Chaudhry, S. A. (2023). An efficient and reliable user access protocol for Internet of Drones. *IEEE Access, 11*, 59688–59700.
63. Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and critical review of RSA based public key cryptographic schemes: Past and present status. *IEEE Access, 9*, 155949–155976.
64. Jin, H., Luo, Y., Li, P., & Mathew, J. (2019). A review of secure and privacy-preserving medical data sharing. *IEEE Access, 7*, 61656–61669.
65. Jose, D. V., & Vijyalakshmi, A. (2018). An overview of security in internet of things. *Procedia Computer Science, 143*, 744–748.
66. Kamers, A. B. (2023). *Homomorphic encryption: Introduction and applicabilities*.
67. Kong, X., Wang, J., Hu, Z., He, Y., Zhao, X., & Shen, G. (2024). Mobile trajectory anomaly detection: Taxonomy, methodology, challenges, and directions. *IEEE Internet of Things Journal*.
68. Koleppusamy, P. *(Note: There is no reference by "Koleppusamy, P." in your provided list. If this is an error or a missing reference, please check your source.)*
69. Kuppuswamy, P., Al, S. Q. Y. A. K., John, R., Haseebuddin, M., & Meeran, A. A. S. (2023). A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm. *Bulletin of Electrical Engineering and Informatics, 12*(2), 1148–1158.
70. Kwame, A. E., Martey, E. M., & Chris, A. G. (2017). Qualitative assessment of compiled, interpreted and hybrid programming languages. Communications on Applied Electronics, 7(7), 8-13.
71. Lin, P. Y., Chang, Y. F., Chang, P. S., & Tai, W. L. (2023, July). Comments on a double-blockchain assisted data aggregation scheme for fog-enabled smart grid. In *International Conference on Frontier Computing* (pp. 234–245). Singapore: Springer Nature Singapore.
72. Liu, P., Xu, X., & Wang, W. (2022). Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives. *Cybersecurity, 5*(1), 4.
73. Mahapatra, B., Krishnamurthi, R., & Nayyar, A. (2019). Healthcare models and algorithms for privacy and security in healthcare records. In *Security and Privacy of Electronic Healthcare Records: Concepts, Paradigms and Solutions* (p. 183).
74. Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H., & Aaraj, N. (2022). Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE, 110*(10), 1572–1609.
75. Mollakuqe, E., Parduzi, A., Rexhepi, S., Dimitrova, V., Jakupi, S., Muharremi, R., ... & Qarkaxhija, J. (2024). Applications of homomorphic encryption in secure computation. *Open Research Europe, 4*(158), 158.
76. Munjal, K., & Bhatia, R. (2023). A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex & Intelligent Systems, 9*(4), 3759–3786.
77. Nazeer, M. I., Mallah, G. A., Shaikh, N. A., Bhatra, R., Memon, R. A., & Mangrio, M. I. (2018). Implication of genetic algorithm in cryptography to enhance security. *International Journal of Advanced Computer Science and Applications, 9*(6), 375–379.
78. Nita, S. L., & Mihailescu, M. I. (2023). *Advances to homomorphic and searchable encryption* (pp. 1–136). Springer.
79. Obaid, T. S. (2020). Study a public key in RSA algorithm. *European Journal of Engineering and Technology Research, 5*(4), 395–398.
80. Ogborigbo, J. C., Sobowale, O. S., Amienwalen, E. I., Owoade, Y., Samson, A. T., & Egerson, J. (2024). Strategic integration of cyber security in business intelligence systems for data protection and competitive advantage. *World Journal of Advanced Research and Reviews, 23*(1), 081–096.
81. Opoku-Mensah, E., Abilimi, C. A., & Boateng, F. O. (2013). Comparative analysis of efficiency of fibonacci random number generator algorithm and gaussian Random Number Generator Algorithm in a cryptographic system. Comput. Eng. Intell. Syst, 4, 50-57.
82. Opoku-Mensah, E., Abilimi, A. C., & Amoako, L. (2013). The Imperative Information Security Management System Measures In the Public Sectors of Ghana. A Case Study of the Ghana Audit Service. International Journal on Computer Science and Engineering (IJCSE), 760-769.
83. Patel, S. (2023). *Evaluating the use of homomorphic encryption for secure data processing in cloud networks* (Doctoral dissertation, Dublin, National College of Ireland).
84. Pearson, S. D. (2020). *Encryption methods in protecting cloud data when adopting cloud solutions: A Delphi study* (Doctoral dissertation, Capella University).
85. Ragavan, M., & Prabu, K. (2019). Dynamic key generation for cryptographic process using genetic algorithm. *International Journal of Computer Science and Information Security (IJCSIS), 17*(4), 246–250.
86. Ragavan, M., & Prabu, K. (2022). Evaluation of cryptographic key generation performance using evolutionary algorithm. *International Journal of System Assurance Engineering and Management, 13*(Suppl 1), 481–487.
87. Sawant, A. S. (2022). *Enhancing encryption in cloud computing and reducing energy usage by using PSO-ALO algorithm to improve homomorphic encryption technique* (Doctoral dissertation, Dublin, National College of Ireland).
88. Seid, A. D. (2024). Privacy preserving biometrics authentication in IoT devices using homomorphic encryption.
89. Sasikumar, K., & Nagarajan, S. (2024). Comprehensive review and analysis of cryptography techniques in cloud computing. *IEEE Access*.
90. Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2022). Integrating encryption techniques for secure data storage in the cloud. *Transactions on Emerging Telecommunications Technologies, 33*(4), e4108.
91. Shankar, K., & Lakshmanaprabu, S. K. (2018). Optimal key based homomorphic encryption for color image security aid of ant lion optimization algorithm. *International Journal of Engineering & Technology, 7*(9), 22–27.
92. Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2024). Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, 1–18.
93. Sun, P. J. (2019). Privacy protection and data security in cloud computing: A survey, challenges, and solutions. *IEEE Access, 7*, 147420–147452.
94. Taherdoost, H., Le, T. V., & Slimani, K. (2025). Cryptographic techniques in artificial intelligence security: A bibliometric review. *Cryptography, 9*(1), 17.
95. Tekin, E. N. (2023). *Homomorphic encryption: A comprehensive study of types, techniques, and real-world applications* (Master's thesis, Middle East Technical University [Turkey]).
96. Thakur, I., Karmakar, A., Li, C., & Preneel, B. (2025). A survey on transciphering and symmetric ciphers for homomorphic encryption. *Cryptology ePrint Archive*.
97. Tyagi, A. K. (Ed.). (2023). *Privacy preservation and secured data storage in cloud computing*. IGI Global.
98. Wardana, A. A., & Sukarno, P. (2024). Taxonomy and survey of collaborative intrusion detection system using federated learning. *ACM Computing Surveys, 57*(4), 1–36.
99. Wood, A., Najarian, K., & Kahrobaei, D. (2020). Homomorphic encryption for machine learning in medicine and bioinformatics. *ACM Computing Surveys (CSUR), 53*(4), 1–35.
100. Xie, Q., Jiang, S., Jiang, L., Huang, Y., Zhao, Z., Khan, S., ... & Wu, K. (2024). Efficiency optimization techniques in privacy-preserving

federated learning with homomorphic encryption: A brief survey. *IEEE Internet of Things Journal*, *11*(14), 24569–24580.
101. Yan, Y. (2022, December). The overview of elliptic curve cryptography (ECC). In *Journal of Physics: Conference Series* (Vol. 2386, No. 1, p. 012019). IOP Publishing.
102. Yeboah, T., Opoku-Mensah, E., & Abilimi, C.A. (2013a). A Proposed Multiple Scan Biometric-Based Registration System for Ghana Electoral Commission. Journal of Engineering, Computers & Applied Sciences (JEC&AS), 2(7).
103. Yeboah, D. T., Odabi, I., & Abilimi Odabi, M. C. A. A. (2016). Utilizing divisible load scheduling theorem in round robin algorithm for load balancing in cloud environment.
104. Yeboah, T., Opoku-Mensah, E., & Abilimi, C. A. (2013b). Automatic Biometric Student Attendance System: A Case Study Christian Service University College. Journal of Engineering Computers & Applied Sciences, 2(6), 117-121.
105. Yeboah T. & Abilimi C.A. (2013). Using Adobe Captivate to creative Adaptive Learning Environment to address individual learning styles: A Case study Christian Service University, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181,www.ijert.org, "2(11).
106. Yigzaw, K. Y., Olabarriaga, S. D., Michalas, A., Marco-Ruiz, L., Hillen, C., Verginadis, Y., ... & Chomutare, T. (2022). Health data security and privacy: Challenges and solutions for the future. In *Roadmap to Successful Digital Health Ecosystems* (pp. 335–362).
107. Zafir, E. I., Akter, A., Islam, M. N., Hasib, S. A., Islam, T., Sarker, S. K., & Muyeen, S. M. (2024). Enhancing security of Internet of Robotic Things: A review of recent trends, practices, and recommendations with encryption and blockchain techniques. *Internet of Things*, 101357.
108. Zhu, Q., Lin, H., Wan, C., Xie, Y., & Peng, S. (2022). Integrity protection for data aggregation in smart grid. *Security and Communication Networks*, *2022*(1), 2734487.
109. Zhu, Q., Lin, H., Wan, C., Xie, Y., & Peng, S. (2022). Research article: Integrity protection for data aggregation in smart grid.