



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

A Polynomial-Root-Based Lightweight Homomorphic Encryption Scheme to Preserve Confidentiality in a Cloud Environment

Krishnakumar D¹, K. R. Ramkumar², Sonam Mittal³

¹Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India.
E-mail: krishna1007cs.phd20@chitkara.edu.in

²Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India. E-mail: k.ramkumar@chitkara.edu.in

³School of Engineering and Technology, CGC University, Jhanjeri, Mohali, Punjab, India. E-mail: Sonam.mittal1287@gmail.com

Abstract

The computations can be carried out straightaway on the ciphertext while keeping the underlying plaintext confidential throughout the process with the help of Fully Homomorphic Encryption [FHE]. This study aimed to develop an enhanced FHE framework using polynomial structures and implant techniques to maintain mathematical accuracy with fewer processing requirements. The novel design incorporates three major components: secret key generation based on polynomial matrices, public key masking using numeric attributes, and a scaling mechanism to standardize and merge the hidden information. Together, these components offer a structured approach to perform operations on protected data while preserving its privacy. The algebraic structure of the polynomial is designed with security aspects that improve operational efficiency in the proposed techniques. By creating a direct linkage between polynomial structure and the encryption process, data confidentiality is maintained as the secret keys are generated from the defined polynomial relationships, which diminishes unnecessary transformation steps and provides a practical alternative to orthodox homomorphic encryption approaches. The accuracy of homomorphic computations is maintained by calculating the scaling factor from the ratio of the aggregated polynomial coefficients to the corresponding aggregated ciphertext results. This normalization helps to regulate the increase in ciphertext values during computation and keeps the encrypted results reliable with the intended plaintext operations without adding substantial computational challenges.

Keywords: Distributed scheming, Data preservation, Fully Homomorphic Encryption, FHE-modeled algorithms, RLWE, NTRU, Polynomial roots and structured algebraic hardness.

1. Introduction

Traditionally, we need to decrypt the original plaintext to perform calculations. Now, the operations are carried out directly on the ciphertext while keeping the underlying plaintext encrypted. This paper focuses on a well-organised approach suggested to perform scalar-based homomorphic calculations, with a focus on weighted addition and multiplication to make encrypted data aggregation more practical for privacy-critical requests where confidentiality and well-organised processing are imperative. In the proposed technique, secret keys are generated from the predefined mathematical constructions to encrypt the plaintext value. On the other end, during decryption, with low computational effort, the original plaintext is recovered using the corresponding mapping structure in a systematic manner. This method is opposite to existing FHE schemes, which generally use secret keys that are randomly generated. In the proposed technique, the secret keys are generated from the pre-stored polynomial values in a structured table using large random prime integers. This idea creates a direct mapping between key generation and the encryption process to provide a reliable mathematical methodology to perform highly secure homomorphic calculations.

During homomorphic operations, to adjust the aggregated weighted values according to predefined reference indexed values, normalization techniques are used to maintain accuracy. To make it even harder from a security standpoint, two stages of XOR-based masking are done by merging the plaintext with secret and public key structures. Additionally, one more layer of complication is introduced using floating-point errors that may be introduced during computations, which are also taken care of through a hexadecimal mapping mechanism. The operations are performed on this multi-layered security mechanism, which adds more complexity and protects the encrypted data from intruders while they try to recover the plaintext.

The proposed scheme has been evaluated using multiple parameters like key, plaintext, and ciphertext sizes, along with execution time in milliseconds and memory utilisation for key generation, encryption, decryption, and complete processing across different dataset sizes.

2. Advancement of Fully Homomorphic Encryption Approaches

In the field of cryptography pertaining to Fully Homomorphic Encryption (FHE), the very first practical construction by Craig Gentry's is considered a significant breakthrough to protect data. The model was designed to demonstrate that arbitrary computations can be carried out on the protected data without exposing the original plaintext. However, there have been many significant challenges, such as computational resource requirements that allow only very limited real-time usage. Over the period, this research domain has progressed well using lattice-based cryptographic techniques, particularly Learning with Errors (LWE) and Ring-LWE (RLWE), focusing on improving processing efficiency and security implications. Due to significant advancements in designing and developing more efficient FHE models, researchers were able to find optimised ways to control ciphertext growth and reduce operational cost for FHE calculations.

These advancements progressed much faster during third-generation FHE schemes, including GSW, and other special schemes such as FHEW and TFHE. The major advancement during this era was improving bootstrapping efficiency and lowering the time complexity for refreshing ciphertexts, which enabled researchers to take a more pragmatic approach to encrypted computation. The CKKS scheme is another significant milestone that plays a vital role in machine learning and data analytics in this domain, which supports approximate arithmetic over real and complex numbers. However, the downstream impact of this scheme pertains to numerical accuracy, operational cost, and compromised security mechanisms, which are areas for researchers to explore in future research. As of today, technological advancements, specifically in cloud platforms, will enable researchers with greater opportunity to optimise and redesign the FHE schemes to preserve and protect sensitive data from intruders during processing.

3. Literature Review

The different studies carried out by the researchers in the FHE models include normalization and optimisation for data aggregation and resource allocation, weighted sums, and linear scaling. Therefore, we need to tweak these FHE models to retain the original value in such a way that the output must satisfy a specific target, for which additional computation is essential. However, the refined techniques will provide only constraint-based solutions due to the higher cost involved for complex and large datasets. These gaps are potential areas where the researchers can do a deep dive as part of future scope of work.

Craig Gentry (2009). The author introduced the first fully homomorphic encryption scheme that allows arbitrary computations to be performed directly on encrypted data based on ideal lattice constructions and enables both addition and multiplication operations. Stehlé, Damien, et al. (2010). The authors give a detailed analysis of one of the hardness assumptions; they discuss the implementation details of an algebraic circuit of low multiplicative degree for a faster FHE scheme. Van Dijk, Marten, et al (2010). The authors constructed an uncomplicated FHE scheme that uses addition and multiplication over integers. They investigated the hardness and the efficiency of the scheme, which can be further improved. Gentry, C., & Halevi, S. (2011, May). The authors aimed to optimize a key-generation technique for SHE; this does not need full polynomial reversal, reducing the time from many hours to days to seconds to minutes. Gentry, C., Halevi, S., & Smart, N. P. (2012, April). The authors gift a construction of FHE schemes that, for a polylog security parameter, will apply to "packed" ciphertexts that write in code vectors of plaintext components with t gates in time t . Conjointly introduced other optimizations that may speed up homomorphic analysis to boost plaintext components to powers of p at the "cost" of a linear operation.

Gentry, C., Sahai, A., & Waters, B. (2013, August). The author proposes a state-of-the-art technique for building FHE schemes that we call the approximate eigenvector method. FHE sum and product are simply matrix operations and multiplication. This makes our theme asymptotically quicker and simpler to grasp. Brakerski, Z., & Vaikuntanathan, V. (2014). The authors present a levelled FHE "somewhat homomorphic" scheme that is purely based on the LWE assumption using a novel re-linearization process. On the other hand, all previous schemes relied on complex guesses to ideals in various rings. Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). The authors present a novel approach to FHE that significantly improves efficiency and bases security on poor assumptions. It is capable of assessing random polynomial-size circuits without using Gentry's rebootable procedure.

Smart, N. P., & Vercauteren, F. (2014). The author proposes SIMD operations with four options. It operates using standard bit-wise FHE schemes and finite field elements to perform SIMD multiplication.

Later, the options were extended to operate on bits and ciphertext to compare Gentry's work with the runtime cost. Brakerski, Z., & Vaikuntanathan, V. (2014, January). The authors show that levelled FHE can be based on the hardness and approximation of lattice problems under classical quantum reductions. This was supported by noise-bounded serial analysis of high fan-in manipulations and, lastly, ordered dimension modulus trimming. Alperin-Sheriff, J., & Peikert, C. (2014, August). The authors gave a new rebooting algorithm whose execution time and associated approximation factor are both small polynomials. Ducas, L., & Micciancio, D. (2015, April). The authors gift a brand-new methodology to homomorphically reckon easy bit operations and bootstrap the ensuing output, which runs on a private pc in exactly half a second. It aims to remove the bottleneck affecting the efficiency of rebooting procedures, to refresh screeching ciphertexts, and to keep manipulations on encrypted information. Chillotti, I., Gama, N., Georgieva, M., & Izabachene, M. (2016, December). A temporal rebooting procedure to reduce the dimensions of keys to keep a minimum identical security is proposed by these authors. Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017, December). The authors propose a new batching technique for an RLWE-based construction. The bit size of the ciphertext modulus grows linearly with the depth of the circuit being evaluated because of the rescaling procedure; the authors aim to beat the problem by remodeling the associated input ciphertext into associated coding of just about constant plaintext with a far larger modulus.

Gentry, C., & Halevi, S. (2019, December). The author provides an additively HE on ring-based LWE with a nearly optimal range. It describes how to compress several FHE ciphertexts, which will have come back from a homomorphic analysis theme, into fewer high-rate ciphertexts for victimisation single-server non-public info retrieval PIR. The server needs to perform a minimum of one bit operation per information bit. El Makkaoui, K., Beni-Hssane et al. (2019) proposed two schemes to speed up the decryption process. The decryption speed-up factors of different variants over one scheme and the other scheme provide faster decryption by increasing the bit numbers. Zuber, M., Carпов, S., & Sirdey, R. (2020, August). The authors investigated two different approaches to classify a transparent storage as returning from one amongst an AN whimsical range of encrypted speaker embeddings; however, it includes a quadratic time-wise complexity within the range of models. The opposite one is matched; however, it is time-linear within the range of the model's victimization SNN computations through the employment of FHE. Alaya, B., Laouamer et al. (2020) review other domains of homomorphic coding and provide a final contrast between the adopted approach. Besides, to support the strongest theme, we should always take into consideration such a large number of parameters in each operation, either coding or cryptography, or the key creation. Zhou, S., Yu, Z., Nasr et al. (2020) propose, because of the low entropy of the matrix notation rule, to boost its entropy. Victimization matrix notation aims to make it appropriate for homomorphic coding schemes over integers, which are rising within the cryptography space to avoid a brute-force attack.

Alabdulatif, A., Khalil, I., & Yi, X. (2020) The authors experiment with the cloud service supplier not having access to cipher information and might perform analysis computations while not having the ability to reveal the content of the initial information. Chillotti, I., Gama et al. (2020) The authors proposed two strategies to control stuffed information in evened-out homomorphic mode, to diminish the ciphertext development, and to improve the assessment of query tables and subjective capacities in RingGSW-based homomorphic plans TBSR. Yousuf, H., Lahzi, M et al. (2021) acquired knowledge of the FHE plot and tracked the best and most recent ways to deal with it and carried them out. After this year, various methodologies like FHE dependent on the Integer and scheme, and the Multikey FHE Scheme dependent on GSW were followed to improve the plan. Caudhari, A., & Bansode, R. (2021). The authors centre around the GSW concept, which is utilized to perform the calculation of scrambled information, gather information from IoT gadgets, and encode it utilizing GSW calculation as an expansion and augmentation activity. Calculation will be performed once the inquiry is terminated on the cloud.

Kalyani, P., Masooda, M., & Namrata, P. (2021). The authors proposed that elliptic curve cryptography (ECC) is utilized for homomorphic encryption of information by taking up moderately less space for capacity. This encryption permits clients to process effectively on the scrambled information without the need to decode it. Jung, W., Kim, S., Ahn, J. H., Cheon, J. H., & Lee, Y. (2021). The authors demonstrated the first GPU implementation for rebooting CKKS. It has been shown that the primary-memory frequency hindrance is the key challenge in accelerating FHE operations with GPUs. It achieved 8 per-slot reboot time with 19-bit exactness, which corresponds to a 257× speedup over the individual-threaded CPU implementation by applying batching plaintext and ciphertext MAC in bootstrapping [55]. Gorantala, S., Springer, R., et al. (2021). The author believes that the XLS intermediate illustration will facilitate reasoning regarding and implementing numerous optimizations. The transpiler presently uses bitwise operations; thus, all arithmetic operations are reduced to several single-bit mathematical gates, making them quite slow. The interpreter within the FHE IR Translation stage provides additional flexibility in execution ways, together with multicore dispatch for improved performance.

Yousuf, H., Lahzi, M., Salloum, S. A., & Shaalan, K. (2021). The author conducts a methodical review through a structured approach; exploration questionnaires are utilized to identify relevant keywords for methodical

assessment of Fully Homomorphic Encryption techniques is carried out in this study. Later, this has been applied to search peer-reviewed journals, conference papers, and scholarly books across established academic catalogues.

Gürsoy, G., Chielle, E., Brannon, C. M., Maniatakos, M., & Gerstein, M. (2022). The authors use a KNN approach to find missing genotypes in a genomic block and implement FHE to avoid decryption of the private genotypes. Mittal, Sonam, Shagun Sharma, and K. R. Ramkumar (2022). The authors proposed an enhanced homomorphic encryption algorithm designed specifically for cloud computing environments, with a focus on improving performance. The approach employs a prime number-based matrix structure for both encryption and decryption, resulting in reduced overall execution time. Rupa, C., Greeshmanth, & Shah, M. A. (2023). The authors proposed a homomorphic encryption technique in plaintext with a primary focus on security challenges on the cloud platform and rapid technological advancement. It is more powerful than the existing encryption methods and resistant to a variety of attacks. This means that an attacker cannot easily predict plaintext through statistical analysis.

Mittal, S., & Ramkumar, K. R. (2023): The core objective of the study is to perform a comparative investigation of data preservation and protection in distributed computation. The conclusion is drawn based on the comparison of relative performance and suitability of different approaches like Fully Homomorphic Encryption (FHE) schemes based on lattice constructions, integer arithmetic, Learning with Errors (LWE), Ring Learning with Errors (RLWE), and Nth-degree Truncated Polynomial Ring Units (NTRU). Boudgoust, K., & Scholl, P. (2023). The authors proposed a prototype implementation within the PALISADE/Open FHE open-source HE library with optimized parameter sets and reported implementation results in terms of runtime and key size with smaller noise growth. Manulis, M., & Nguyen, J. (2024, May). The author introduced a new security notion known as security against verified chosen-ciphertext attacks (vCCA). Two formal definitions were presented: one based on the indistinguishability framework and the other following a simulation-based non-malleability approach, which extends the concept of targeted malleability.

Wang, X., & Li, P. (2024). The author proposed a secure and privacy-preserving face verification scheme that employs Fully Homomorphic Encryption and SealPIR. The approach follows a three-party architecture, where an additional server trusted by the client is used to support anonymous verification. Valera-Rodriguez, F. J., Manzanera-Lopez, P., & Cano, M. D. (2024). The author resolves privacy risks in cloud environments; the author conducts a wide-ranging assessment of Fully Homomorphic Encryption (FHE) techniques, converging on practical distributions using the Microsoft SEAL library.

4. Polynomial and Root-Based Key Generation Construction

The proposed homomorphic encryption technique employs an organised polynomial outline for secret key generation. The quadratic polynomials are formed using arbitrary and algebraic construction by selecting the prime coefficients using a random method. The framework proposes a real-time balance between operational efficiency and security, arranged using a hexadecimal matrix table, in which the respective matrix

spot (i, j) is connected with a certain quadratic polynomial, in which the second-degree polynomials are plotted based on the random index values, and the identified polynomials are used to find their root value pairs to generate the secret keys. Plotting the original polynomials and their roots is very difficult, as the logic is designed in a simpler form that enables efficient computation with very limited information.

4.1. Structure of the Polynomial Matrix

The coefficient architecture is defined by a hexadecimal matrix table, where each distinct component is represented by a quadratic polynomial of the form: $P_{ij}(x) = a_{ij}x^2 + b_{ij}x + c_{ij}$, then outline $R(P_{ij}) = \{r_1, r_2\}$ and $S = A(R, C)$, where R and C are catalogue sets arbitrarily selected as prime coefficients. Mapping a unique quadratic polynomial to each element of the matrix will create secret key generation that is possibly a larger set. A balance between key diversity and operational cost was added to the polynomial structures with variation using randomly selected prime coefficients. These architectural orchestrations are designed to support efficient key generation to help in mitigating the significant processing overhead.

4.2. Submatrix Selection and Root Extraction

In the selected key generation method, the positions are chosen randomly from the coefficient matrix to select the quadratic polynomials, from which the secret key and its roots are derived. As the selection method of row and column positions is kept private, intruders may face difficulty knowing the actual polynomial used. With the available partial ciphertext information, this approach will complicate the process of retrieving the original information. This approach will further complicate the possible selections of the key-search space based on the large prime numbers, which results in maintaining low computational overhead.

Table 1 Polynomial Matrix

Hexadecimal 2 nd degree Polynomial Roots	00	01	02	03	04	05	06	07	08	09	A	B	C	D	E	F
00	x ² -150x+5723	x ² -42x+377	x ² -50x+588	x ² -76x+2003	x ² -53x+235	x ² -62x+817	x ² -108x+2867	x ² -76x+1219	x ² -143x+4171	x ² -18x+187	x ² -92x+851	x ² -126x+3293	x ² -90x+1541	x ² -108x+1691	x ² -72x+1271	x ² -66x+1073
01	x ² -40x+831	x ² -44x+403	x ² -58x+795	x ² -78x+5237	x ² -58x+265	x ² -70x+601	x ² -114x+3149	x ² -82x+1557	x ² -100x+3059	x ² -38x+308	x ² -64x+943	x ² -134x+3589	x ² -94x+1633	x ² -118x+1843	x ² -74x+1333	x ² -70x+1189
02	x ² -46x+493	x ² -50x+481	x ² -4x+4	x ² -5x+8	x ² -64x+295	x ² -7x+10	x ² -118x+3337	x ² -9x+24	x ² -102x+2117	x ² -34x+251	x ² -86x+989	x ² -13x+21	x ² -96x+1679	x ² -15x+26	x ² -78x+1457	x ² -72x+1247
03	x ² -48x+521	x ² -54x+533	x ² -130x+4389	x ² -6x+8	x ² -86x+305	x ² -8x+15	x ² -120x+3431	x ² -58x+21	x ² -188x+2291	x ² -40x+318	x ² -48x+287	x ² -14x+33	x ² -102x+2817	x ² -16x+39	x ² -84x+1645	x ² -76x+1363
04	x ² -54x+629	x ² -56x+559	x ² -132x+4307	x ² -86x+2279	x ² -72x+335	x ² -94x+913	x ² -126x+3713	x ² -108x+2747	x ² -112x+2407	x ² -42x+341	x ² -50x+301	x ² -32x+147	x ² -106x+2909	x ² -126x+1739	x ² -92x+1829	x ² -82x+1537
05	x ² -58x+687	x ² -60x+611	x ² -138x+4661	x ² -102x+2537	x ² -76x+355	x ² -10x+25	x ² -30x+51	x ² -112x+2911	x ² -118x+2581	x ² -48x+407	x ² -54x+329	x ² -16x+55	x ² -112x+2047	x ² -18x+85	x ² -92x+1891	x ² -88x+1713
06	x ² -60x+721	x ² -66x+689	x ² -142x+4897	x ² -104x+2623	x ² -78x+385	x ² -19x+34	x ² -32x+87	x ² -114x+2993	x ² -126x+2813	x ² -52x+451	x ² -82x+271	x ² -130x+3901	x ² -126x+2231	x ² -108x+2067	x ² -88x+2077	x ² -90x+1789
07	x ² -64x+799	x ² -72x+767	x ² -36x+323	x ² -62x+277	x ² -84x+395	x ² -21x+38	x ² -36x+68	x ² -14x+49	x ² -42x+407	x ² -54x+473	x ² -86x+413	x ² -52x+667	x ² -61x+118	x ² -66x+899	x ² -86x+1143	x ² -78x+1517
08	x ² -84x+1263	x ² -74x+793	x ² -72x+1067	x ² -64x+183	x ² -88x+415	x ² -25x+46	x ² -12x+57	x ² -12x+83	x ² -98x+2021	x ² -58x+517	x ² -88x+427	x ² -118x+2883	x ² -63x+122	x ² -122x+1897	x ² -208x+2491	x ² -112x+1277
09	x ² -102x+2021	x ² -80x+871	x ² -78x+1121	x ² -70x+201	x ² -94x+445	x ² -31x+58	x ² -34x+81	x ² -14x+95	x ² -24x+119	x ² -64x+581	x ² -74x+669	x ² -114x+2053	x ² -69x+134	x ² -126x+3569	x ² -80x+1591	x ² -88x+1207
A	x ² -104x+2363	x ² -84x+923	x ² -80x+1159	x ² -74x+213	x ² -102x+485	x ² -33x+62	x ² -46x+111	x ² -18x+115	x ² -26x+133	x ² -70x+648	x ² -78x+497	x ² -116x+3139	x ² -73x+142	x ² -112x+3827	x ² -84x+1789	x ² -98x+1241
B	x ² -110x+2449	x ² -86x+949	x ² -86x+1273	x ² -78x+219	x ² -120x+3551	x ² -39x+74	x ² -44x+123	x ² -34x+145	x ² -30x+161	x ² -72x+671	x ² -80x+511	x ² -22x+121	x ² -75x+146	x ² -24x+343	x ² -92x+1961	x ² -96x+1343
C	x ² -114x+2573	x ² -82x+1027	x ² -90x+1448	x ² -82x+237	x ² -124x+3763	x ² -43x+82	x ² -40x+129	x ² -38x+155	x ² -36x+203	x ² -78x+713	x ² -88x+553	x ² -88x+1927	x ² -81x+158	x ² -116x+4183	x ² -86x+2183	x ² -108x+1411
D	x ² -120x+2759	x ² -86x+1079	x ² -92x+1387	x ² -88x+249	x ² -126x+3869	x ² -45x+86	x ² -50x+141	x ² -42x+185	x ² -38x+217	x ² -82x+781	x ² -90x+581	x ² -94x+2173	x ² -85x+166	x ² -26x+169	x ² -98x+2257	x ² -106x+1513
E	x ² -126x+3007	x ² -102x+1157	x ² -98x+1503	x ² -92x+267	x ² -132x+4187	x ² -49x+94	x ² -56x+159	x ² -48x+205	x ² -44x+258	x ² -84x+803	x ² -98x+623	x ² -100x+2419	x ² -91x+178	x ² -118x+2923	x ² -504x+2479	x ² -114x+1649
F	x ² -138x+3977	x ² -120x+1261	x ² -102x+1577	x ² -100x+291	x ² -138x+4599	x ² -55x+106	x ² -124x+1403	x ² -48x+215	x ² -144x+4559	x ² -90x+868	x ² -104x+679	x ² -102x+2501	x ² -99x+194	x ² -122x+3071	x ² -208x+2627	x ² -150x+5141

4.3. Security Implications

Instead of directly revealing the submatrix positions and polynomial roots, exclusive key retrieval is practically difficult, as the plaintext and ciphertext information offer insufficient information; in this way, the security mechanism is achieved. This offers another security mechanism against conventional lattice-based models that enables upholding low computational cost.

4.4. Role in the Overall Encryption Scheme

The ciphertext growth during homomorphic calculations is controlled by using structured quadratic polynomials in the key generation and lightweight XOR operations for masking, which reduce the computational overhead in the proposed scheme. The complex noise-management manoeuvres are used to reduce the need for repeated transformations, as this method requires far fewer processing steps for encrypted aggregations, encryption, and decryption.

5. Optimized Fully Generic Homomorphic Encryption

The computations can be carried out straightaway on the ciphertext while keeping the underlying plaintext confidential throughout the process with the help of Fully Homomorphic Encryption [FHE]. This was achieved with the help of structured polynomial relationships, as the secret keys are generated as scalar values. Additionally, one more layer is added on top of lightweight XOR masking and ciphertext scaling to safeguard and process the encrypted data. These proposals permit computational exactness and low processing overhead and are very much suitable for real-time privacy-preserving applications.

Let the plaintext statement set be denoted as

$$\mathbf{M} = \{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n\}, \mathbf{M}_i \in \mathbf{Z}$$

In the proposed method, both additive and multiplicative homomorphic operations are achieved directly on ciphertexts by maintaining consistency with the respective plaintext computations being supported. During the computation process, the ciphertext growth is controlled using scaling factors. This approach enables simplifying the overall design, drastically minimising the computational difficulties pertaining to traditional FHE schemes. The secret keys are derived from the basic polynomial construction, and each plaintext value is linked with it in the proposed model. To protect the data during transit and dissemination, scalar multiplication with lightweight XOR masking techniques is used during the encryption process.

5.1. Key Generation Using Polynomial Structures

The quadratic polynomials are derived from the coefficient matrix to produce the secret keys in the proposed model. The equivalent polynomials are selected using random rows and columns to identify the smaller submatrices, in which the root value pairs are obtained. The isolated polynomial structure and the collection

indices are used to increase the number of possible key blends, which makes it difficult to recover the original secret key from the inadequate information. A fixed polynomial coefficient matrix $A \in \mathbb{Z}^{16 \times 16}$, where each matrix component represents a second-degree polynomial. In the proposed scheme, the rows and columns are selected in a random style during key generation, in which the smaller polynomial submatrices are formed from the main matrix. The structured randomised process provides a dependable basis while aiming to balance security, computational efficiency, and real-time implementation.

The root values attained from the nominated quadratic polynomials are then used to arrange the secret key set, $S = \{sk_1, sk_2, \dots, sk_n\}$. The measured arbitrariness was introduced in the proposed model by preserving a clear mathematical association between polynomial construction and the encryption method. By shaping the corresponding roots with limited evidence, intruders cannot identify the polynomials used to generate the secret key, as the selected row and column positions are kept private. In balancing security and computational efficiency, the structured secret key generation and lightweight public key masking were used in the proposed framework. In contrast, integers and suitable factors are selected randomly, while independent public keys are generated. In the homomorphic calculations, the keys are used only for masking during encryption and decryption. This separation provides an additional layer of protection, which reduces the computational overhead.

5.2. Encryption Process

In the proposed encryption framework, each plaintext value (M_i) is increased by its corresponding secret key (sk_i) to yield a midway encrypted value. This operation forms an initial ciphertext and presents a direct connection between the plaintext and the secret key. To further toughen security, the halfway value is concealed using two consecutive XOR masking operations performed with independently generated public keys. The encryption process can be expressed as: $C_i = M_i \cdot sk_i$ where (C_i) represents the transitional ciphertext value. To prevent discovery of this value, two public-key-based masking manoeuvres are applied: $CM_i = C_i \oplus pk_{i1} \oplus pk_{i2}$ where (pk_{i1}) and (pk_{i2}) are independently created public keys used solely for hiding purposes. The XOR hiding affords a modest and computationally efficient method to shield the intermediate ciphertext. Since XOR is reversible, the same keys can be used in reverse order during decryption to recover the original intermediate value. This conserves the required mathematical relationship for additional homomorphic processing.

By merging secret-key multiplication with insubstantial XOR hiding, the proposed scheme provides privacy while supporting both additive and multiplicative operations on encrypted data. The approach allows calculations to be executed without directly revealing the underlying plaintext, making it suitable for privacy-preserving data processing.

5.3. Decryption Process

The decryption process of the proposed Fully Homomorphic Encryption scheme is designed to be simple, efficient, and algebraically consistent with the encryption and homomorphic evaluation procedures. Decryption reverses the masking and scaling operations applied during encryption and evaluation, allowing the correct plaintext result to be recovered without exposing intermediate sensitive information. Decryption of Individual Ciphertexts can be understood as follows:

Let the encrypted ciphertext corresponding to plaintext. M_i be: $CM_i = (M_i \cdot sk_i \oplus pk_{i1}) \oplus pk_{i2}$, where sk_i is the secret scalar key and pk_{i1}, pk_{i2} are public masking keys. Since XOR is an involutive operation, the masking is removed by reapplying the same public keys: $C_i = (CM_i \oplus pk_{i2}) \oplus pk_{i1}$. This yields the intermediate value:

$$C_i = M_i \cdot sk_i.$$

The original plaintext is recovered by dividing out the secret key: $M_i = \frac{C_i}{sk_i}$, provided $sk_i \neq 0$ and arithmetic is performed over an exact or bounded-precision domain. Since secret keys are generated as polynomial roots over a controlled domain, this condition is always satisfied.

Now, discussing the additive homomorphism property. Let the additive homomorphic aggregation of ciphertexts be: $C_{add} = \sum_{i=1}^n C_i M_i$. After removing XOR masking: $C_{add}^* = \sum_{i=1}^n (M_i \cdot sk_i)$. During evaluation, a scaling factor $k_{add} = \frac{P_{add}}{C_{add}^*}$ is applied, where $P_{add} = \sum_{i=1}^n P_i$ is the polynomial-domain aggregate. The scaled ciphertext is therefore: $\tilde{C}_{add} = k_{add} \cdot C_{add}^* = P_{add}$. Since the scaling aligns the ciphertext result with the polynomial coefficient aggregate, decryption is achieved directly by interpreting $\tilde{C}_{add}$ in the polynomial domain.

$$\text{Dec}(C_{\text{add}}) = \sum_{i=1}^n M_i.$$

Similarly, for multiplicative homomorphic evaluation, the ciphertext aggregation is: $C_{\text{mul}} = \prod_{i=1}^n C M_i$. After unmasking: $C_{\text{mul}}^* = \prod_{i=1}^n (M_i \cdot sk_i)$. A scaling factor is computed as: $k_{\text{mul}} = \frac{P_{\text{mul}}}{C_{\text{mul}}^*}$, where $P_{\text{mul}} = \prod_{i=1}^n P_i$. The scaled ciphertext becomes: $\tilde{C}_{\text{mul}} = k_{\text{mul}} \cdot C_{\text{mul}}^* = P_{\text{mul}}$. Decryption is therefore achieved by directly mapping the scaled ciphertext back to the polynomial coefficient product, yielding:

$$\text{Dec}(C_{\text{mul}}) = \prod_{i=1}^n M_i.$$

This process ensures multiplicative correctness while maintaining bounded ciphertext growth.

5.4. Correctness of Proposed Algorithm

The correctness of the decryption process follows from the deterministic cancellation of masking and scaling operations, as follows:

- XOR masking is perfectly reversible due to the involutive property of XOR.
- Secret key multiplication is inverted through division by the corresponding secret key.
- Scaling factors normalize aggregated ciphertexts to match polynomial-domain aggregates.

As a result, the decrypted output exactly corresponds to the intended plaintext operation, provided arithmetic is performed in a domain that preserves division accuracy.

Let ℓM and ℓsk denote the bit-lengths of plaintexts and secret keys, respectively. Each intermediate ciphertext satisfies:

$$\ell C_i = O(\ell M + \ell sk).$$

Additive aggregation increases bit-length logarithmically with the number of ciphertexts, while multiplicative aggregation increases it linearly. The explicit scaling step bounds this growth, ensuring ciphertext sizes remain polynomially bounded for fixed-depth aggregation.

6. Fully Homomorphic Algorithm

The algorithm can be formalised using set-builder notation to describe the process of scaling product values. Each plaintext is associated with a unique scalar secret key derived from a structured polynomial construction. Encryption is performed using scalar multiplication followed by lightweight XOR-based masking. Homomorphic evaluation is supported through aggregation operations, with ciphertext growth controlled by an explicit scaling factor. By supporting both addition and multiplication on encrypted data, the proposed scheme qualifies as a mixed homomorphic encryption system. Arbitrary combinations of these operations can be performed directly on ciphertexts without exposing plaintext values, enabling secure aggregation and computation in encrypted domains. Since the proposed scheme supports both addition and multiplication over encrypted data, it qualifies as a mixed homomorphic encryption scheme. Arbitrary sequences of these operations can be applied directly to ciphertexts without revealing plaintext values. The proposed algorithm given below, a novel optimized polynomial-based Fully Generic Homomorphic Encryption, is designed around four interconnected sub-algorithms that govern the generation of cryptographic keys, secure data transformation, ciphertext processing, and plaintext recovery. So, an accurate product of plaintexts is obtained after decrypting the multiplication of ciphertexts. Hence, the multiplicative homomorphic property is achieved.

Sub-Algorithm1: KeyGen: Key Generation Algorithm

- i. Generate random polynomials $P_i \in [P_1, P_2, \dots, P_n]$ from the 16-by-16 polynomial matrix as P_i and generate the $S_i \in [S_1, S_2, \dots, S_n]$ by selecting the polynomial root value pair by finding its roots
- ii. Select the secret key as S_{sk} , from these root value pairs.
- iii. Select the Public key as P_{pk} , by choosing a random number, find all its factors to select any two public keys from the list.
- iv. Build the plaintext and its associated secret and public keys for $M1: [M_1, sk_1, pk_{11}, pk_{12}]$
- v. Repeat steps i to iii and build the plaintext and associated secret and public keys for $M2: [M_2, sk_2, pk_{21}, pk_{22}]$

Sub-Algorithm2: Encrypt: Encryption Algorithm

- i. Input the plaintext or message as M , where M_1, M_2
- ii. The Cumulative Plaintext operations of M , $[P_{add} \mid P_{mul}]$, are given by $P_{add} = \sum_{i=1}^n Pmi$. $\mid$ $P_{add} = \sum_{i=1}^n Pmi$.
- iii. Generate the ciphertext using CM1 XOR operations:

$$[C_1 = M_1 * sk_1] \rightarrow [C_{x1} = C_1 XOR pk_{11}] \rightarrow [C_{M1} = C_{x1} XOR pk_{12}]$$
- iv. Generate the ciphertext using CM2 XOR operations:

$$[C_2 = M_2 * sk_2] \rightarrow [C_{x2} = C_2 XOR pk_{21}] \rightarrow [C_{M2} = C_{x2} XOR pk_{22}]$$
- v. The Cumulative Ciphertext homomorphic operations of ciphertext $[C_{add} \mid C_{mul}]$ are given by

$$C_{add} = \sum_{i=1}^n Cmi. \mid C_{mul} = \sum_{i=1}^n Cmi.$$

Sub-Algorithm3: Decrypt: Decryption Algorithm

- i. The scaling factor (sf) is calculated as:

$$k = \frac{P_{add}}{C_{add}} \mid k = \frac{P_{mul}}{C_{mul}}$$
- ii. The scaled product set C' is then defined as: $C' = \{Ci' \mid Ci' = Ci * k_i, \forall i \in \{1, 2, \dots, n\}\}$.
- iii. Input C_{add} And multiply it by the scaling factor (k) and obtain the M_{add} .
- iv. Apply the Reverse process (M_{add}) to retrieve the original plaintext, M
- v. Input C_{mul} And multiply it by the scaled factor (k) and obtain the M_{mul} .
- vi. Apply the Reverse process (M_{mul}) to retrieve the original plaintext, M .

Sub-Algorithm4: Eval: Evaluation Algorithm

- i. The scaling factor (sf) is calculated as: $k = \frac{P_{add}}{C_{add}} \mid k = \frac{P_{mul}}{C_{mul}}$
- ii. The scaled product set C' is then defined as: $C' = \{Ci' \mid Ci' = Ci * k_i, \forall i \in \{1, 2, \dots, n\}\}$
- iii. Input C_{add} And multiply it by the scaling factor (k) and obtain the M_{add} .
- iv. Apply the Reverse process (M_{add}) to retrieve the original plaintext, M
- v. Input C_{mul} And multiply it by the scaling factor (k) to obtain the M_{mul} .
- vi. Apply the Reverse process (M_{mul}) to retrieve the original plaintext M .

Table 3: Different property analysis of proposed algorithm vs Other FHE schemes

Scheme	Key Generation	Encryption	Homomorphic Evaluation	Growth Control	Security Assumption
Proposed Scheme	Polynomial root computation	Integer multiplication + XOR	$O(n)$ add, scaled multiply	Scaling normalization	Polynomial-root hardness
BGV	Ring sampling	Ring arithmetic	Deep circuits	Modulus switching	RLWE
BFV	Ring sampling	Ring arithmetic	Leveled FHE	Re-linearization	RLWE
CKKS	Ring sampling	Ring arithmetic	Approximate arithmetic	Rescaling	RLWE (approx.)

7. Result analysis

In this section, the proposed homomorphic encryption scheme is compared with some existing standards, such as Gentry's algorithm, DGHV, LWE, RLWE, and NTRU, to validate the proposed model. The comparisons are done based on metrics such as encryption time, decryption time, and memory usage. The relative analysis has been **completed and publicised with the support of diverse graphs and statistics.**

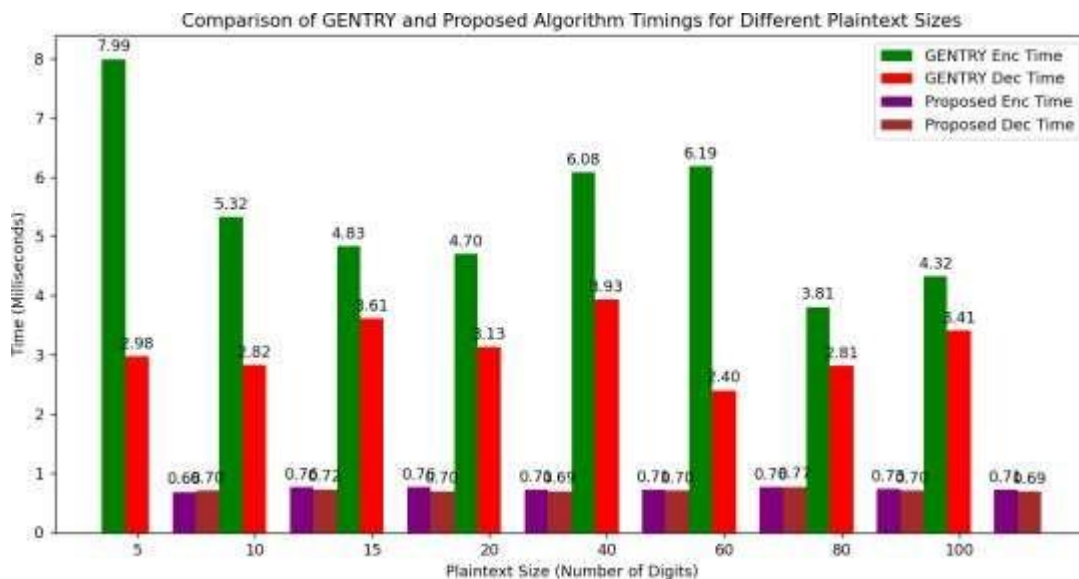
**Figure 5: Time complexity analysis of proposed algorithm vs Gentry**

Figure 5 demonstrates an edge over Gentry's algorithm. While Gentry's approach exhibits higher computational overhead due to its dependence on more complex cryptographic operations, the proposed algorithm significantly reduces both encryption and decryption time. As shown in the comparative analysis, the proposed algorithm scales more efficiently with increasing plaintext sizes, performing better in overall execution time. Also, the key generation and ciphertext size overhead are minimized, further enhancing its effectiveness. Thus, the offered algorithm outperforms Gentry's algorithm in both time complexity and practical interpretation.

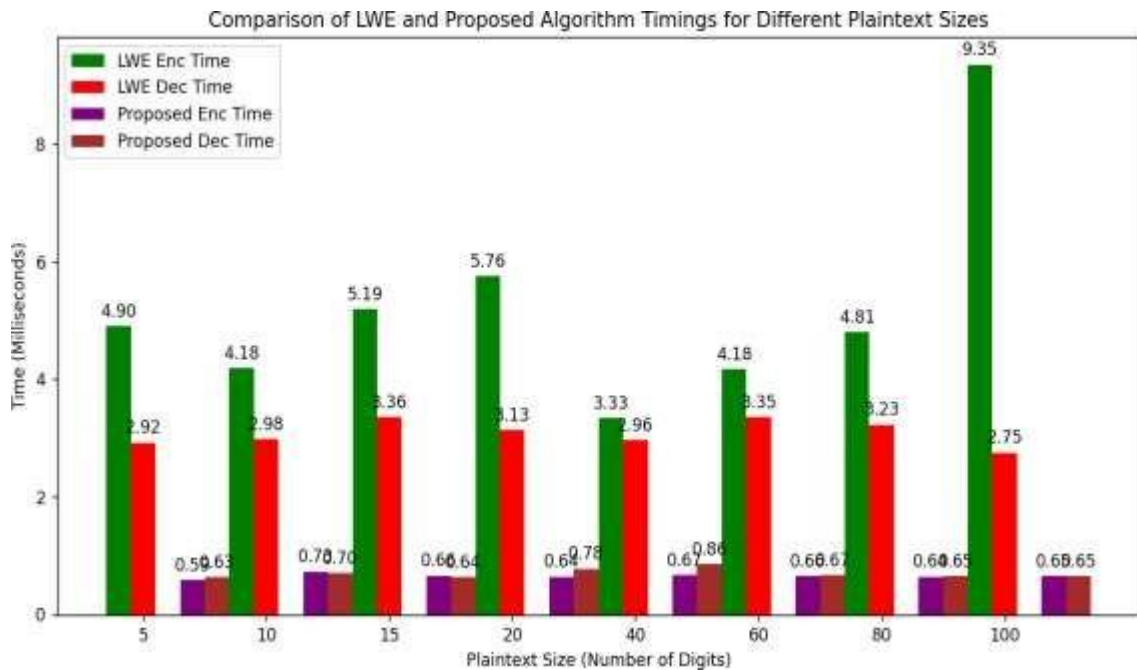


Figure 6: Time complexity analysis of proposed algorithm vs LWE

In Figure 6, the interpretation comparison between the proposed encryption algorithm and the LWE scheme demonstrates that the proposed system significantly reduces both encryption and decryption time complexity. As shown in the graph, the decryption outperforms LWE by a considerable margin. The proposed encryption scheme maintains stable encryption performance even as the size of the plaintext increases.

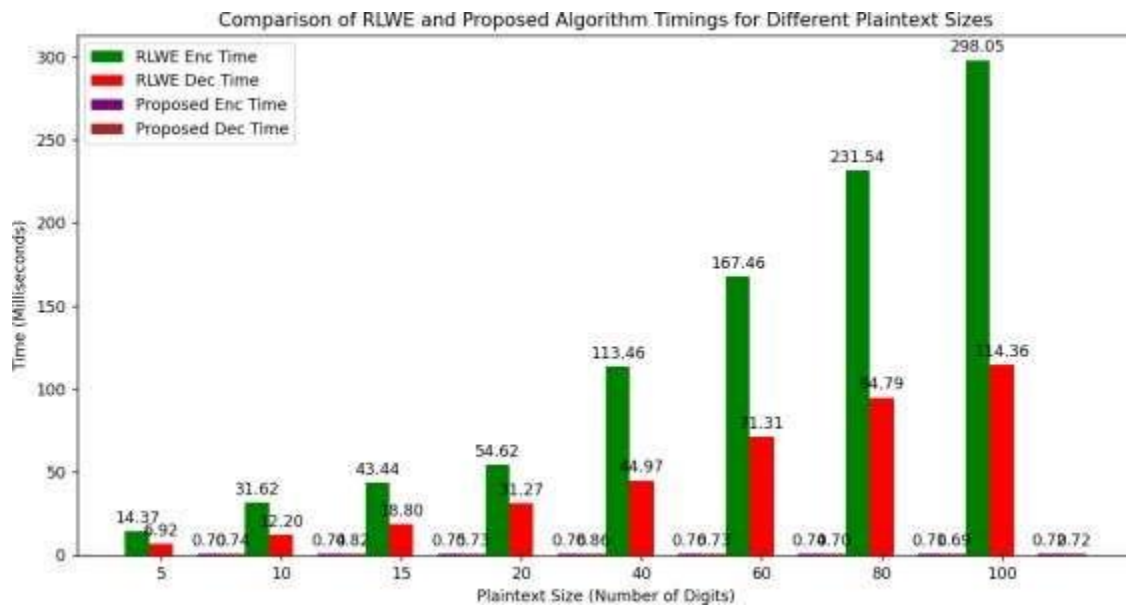


Figure 7: Time complexity analysis of proposed algorithm vs RLWE

Figure 7, compares the proposed method against RLWE; it performs relatively better as plaintext size increases, with lower and more consistent execution time for both encryption and decryption. Also, it is more secure, requires less runtime, has reduced computational overhead, and the performance is stable during data processing.

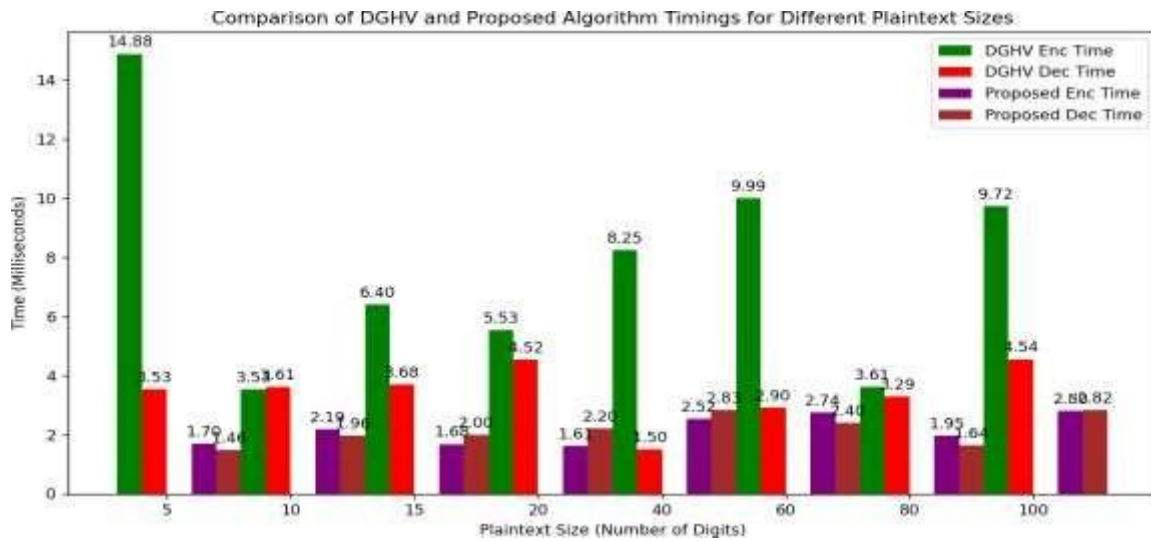


Figure 8: Time complexity analysis of proposed algorithm vs DGHV

In Figure 8, the interpretation compares the proposed method against DGHV; it performs relatively better as plaintext size increases, with lower and more consistent execution time for both encryption and decryption. Also, it is more secure, requires less runtime, has reduced computational overhead, and the performance is stable during data processing.

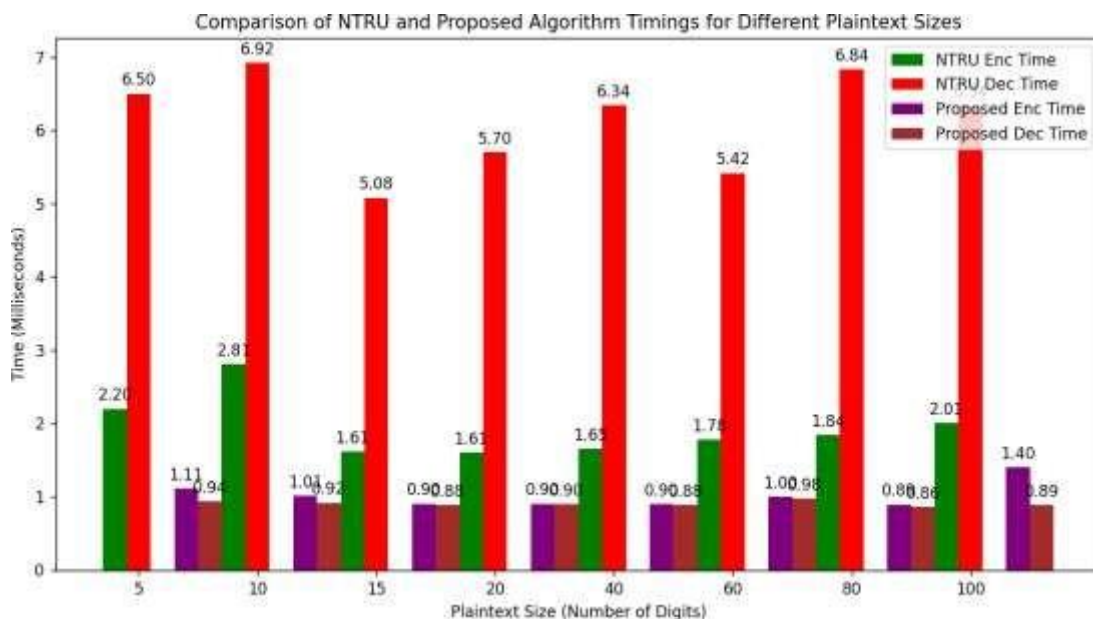


Figure 9: Time complexity analysis of proposed algorithm vs NTRU

In Figure 9, the interpretation compares the proposed method against NTRU; it performs relatively better as plaintext size increases, with lower and more consistent execution time for both encryption and decryption. Also, it is more secure, requires less runtime, has reduced computational overhead, and the performance is stable during data processing.[64]. A cumulative analysis of the earlier specified metrics was done. Gentry's, DGHV, LWE, RLWE, NTRU, and the proposed FHE are compared with the proposed model of an enhanced homomorphic encryption scheme.

Table 4 shows the numerical and relative comparison of the proposed model variants and existing standards. The comparison is made for the plaintext of different-length digits, i.e., the plaintext is an integer of different lengths of digits. The key generation time, encryption time, decryption time, and total duration for the execution of the complete algorithm are given in milliseconds (ms). The lengths of the ciphertext and secret key are measured

in digits, i.e., the number of digits in the ciphertext and secret key, respectively, as well as in bytes. The total memory required for the execution of the different algorithms is also measured in bytes.

Table 4. Relative Comparative Analysis of Different Existing Standards with Proposed Model

Parameters	DGHV	Gentry	LWE	RLWE	NTRU	Proposed
Plaintext Size (Bytes)	12856	12856	12856	12856	12856	72
Ciphertext Size (Bytes)	12856	12904	25704	409736	832	72
Secret Key Size (Bytes)	28	60	124	696	96	84
Memory Usage (Bytes)	6870	7301	7653	9537	8890	10785 (7096)
Key Generation Time (ms)	47.153	0.9287	0.8304	1.274	3521.06	1.177
Encryption Time (ms)	44.764	27.0298	33.6856	479.735	1.808	0.977
Decryption Time (ms)	17.170	16.592	12.2163	209.494	10.656	1.146
Total Duration (ms)	109.087	44.551	46.7323	690.503	3533.533	3.301

The relative comparison of existing standards with variants of the proposed model in Table 5 shows that the proposed model is more efficient in terms of overall execution time. The growth in ciphertext size is also less than that of lattice-based (Gentry's algorithm), integer-based (DGHV), finite field-based (LWE and RLWE), NTRU FHE, and Camouflage FHE schemes. The secret key size is also smaller than that of other algorithms, except DGHV, Gentry, and NTRU. The memory required is also the least in the case of the proposed algorithm after subtracting the memory required to store the generated polynomial matrix, except for the integer-based DGHV scheme.

Table 5. 15 Relative Comparative Analysis of Different Existing Standards and Variants of the Proposed Model

Parameters	DGHV	Gentry	LWE	RLWE	NTRU	Proposed
Ciphertext Size	Very Large	Very Large	Very Large	Very Large	Very Large	Comparable
Secret Key Size	Small	Small	Large	Large	Very Large	Small
Memory Usage	Less	Comparable	More	More	More	More
Key Generation Time	Very Large	Less	Less	Comparable	Very Large	Large
Encryption Time	Large	Large	Large	Very Large	Comparable	Large
Decryption Time	More	Less	More	Very Large	More	More
Total Duration	Large	More	More	Very Large	Very Large	More

7.1. Discussion

This section summarizes the results and validation for the proposed model and shows the cumulative analysis of different existing fully homomorphic standards with the proposed model. The different tables help to summarize all the results and comparative analysis of all the work done in this research. It also summarizes the parameters on which the proposed algorithm is evaluated. The results show that the proposed FHE model, an enhanced homomorphic encryption model for preserving privacy, performs better than existing standards of FHE models in the cloud environment.

The proposed model introduces a polynomial-based FHE scheme that makes use of random prime numbers, matrix operations, and randomization techniques to improve privacy protection in cloud environments. It works with a fixed 16×16 matrix of quadratic polynomials having randomly chosen coefficients. The secret keys are derived by selecting hidden row and column submatrices and extracting the roots of the polynomials, whereas the public keys are obtained from the factors of random integers and are used only for masking purposes. The performance of the model was evaluated on several parameters, namely key generation, encryption and decryption time, total execution, computational time, ciphertext length, secret-key size, and memory usage. The results show that key generation, encryption, and decryption times stay in the range of about 0.6 to 1.4 ms for plaintext sizes from 5 to 100 digits, which indicates that the scheme scales efficiently and does not carry the high per-operation cost usually seen in lattice-based FHE. The ciphertext grows almost in proportion to the number of plaintext digits, from 28 bytes for a 5-digit plaintext to 72 bytes for a 100-digit plaintext, so the expansion is linear, modest, and well controlled. Total memory consumption remains constant at 10,785 bytes for all tested plaintext lengths, and the key size is fixed at 84 bytes throughout. This shows that the memory footprint is mainly decided by fixed overhead rather than by the input size, and most of this memory is used by the 16×16 matrix of randomly chosen quadratic polynomials.

When compared with existing FHE schemes, the proposed model clearly shows better computational efficiency. Some existing schemes, particularly Gentry, LWE, and RLWE, may need less time for key generation, but their encryption and decryption operations take much more time. The proposed model consistently achieves lower encryption, decryption, and total execution times than all five compared schemes. Its key generation time is comparable to Gentry and LWE, faster than DGHV, and dramatically faster than NTRU. The ciphertext and secret key sizes are also significantly smaller than those of Gentry, DGHV, LWE, RLWE, and NTRU for all plaintext sizes, with lower ciphertext expansion as the plaintext size increases. In terms of space complexity, the proposed model performs better than Gentry and DGHV, but its total memory footprint is slightly higher than LWE, RLWE, and NTRU, although it stays stable across all plaintext sizes. In other words, the scheme accepts a small increase in memory compared to these newer lattice-based schemes in return for large gains in computational speed and in the compactness of ciphertexts and keys, making it competitive on almost every metric except raw memory usage.

The study also gives an extensive experimental comparison with established FHE approaches, including Gentry, DGHV, LWE, RLWE, and NTRU, which provides a proper assessment of the proposed model on both computational and storage-related parameters. Overall, the proposed model shows improved time and space efficiency across the reported experimental parameters, while still maintaining the required homomorphic-encryption functionality, and it offers a good alternative approach for privacy-preserving computation in cloud environments.

8. Conclusion

The proposed FHE technique is designed on the basis of a well-defined algebraic construction; therefore, it is very difficult to break from a computational perspective. The complexity of this scheme relies on the secret keys being generated from polynomial root values that are present in the submatrices, which are in the range of a larger matrix of polynomial coefficients. The specific rows and columns used in key generation are not exposed. An attacker endeavouring to recover the secret key would need to correctly identify the underlying polynomial substructure and reconstruct valid root information using only partial ciphertext data. Even though polynomial factorisation and root computation can be achieved proficiently when wide-ranging information is available, the contest in this scheme comes from the large number of possible groupings and the lack of complete structural data, rather than from classical NP-hard problems. This study presents a detailed examination of FHE and reviews the major developments and improvements that have contributed to its evolution. Privacy preservation is achieved in many applications such as financial analytics, cloud data processing, and optimisation, using a scaling mechanism that provides a pragmatic framework that preserves proportionality during weighted homomorphic aggregation. By exploring nonlinear optimisation models, higher-degree polynomials and dynamic scaling can be future research scope.

References

1. Gentry, C. (2009), Fully homomorphic encryption using ideal lattices". In Proceedings of the forty-first annual ACM symposium on Theory of Computing, Association for Computing Machinery, New York, NY, United States. pp. 169-178.
2. Stehlé, Damien, and Ron Steinfeld. "Faster fully homomorphic encryption." In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2010.
3. Gentry, Craig, Shai Halevi, and Nigel P. Smart. "Better bootstrapping in fully homomorphic encryption." In proceedings of the International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2012.

4. Van Dijk, Marten. (2010), Fully homomorphic encryption over the integers." In Proceedings of the Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg. Pp. 24-43.
5. Gentry, C., & Halevi, S. (2011). Implementing gentry's fully-homomorphic encryption scheme. In Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, pp. 129-148.
6. Gentry, Craig, and Shai Halevi. "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits." In proceedings of IEEE 52nd annual symposium on foundations of computer science. IEEE, 2011.
7. Brakerski, Zvika, and Vinod Vaikuntanathan. "Fully homomorphic encryption from ring-LWE and security for key-dependent messages." In Proceedings of the Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2011.
8. Brakerski, Zvika. "Fully homomorphic encryption without modulus switching from classical GapSVP." In proceedings of Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2012.
9. Coron, Jean-Sébastien, David Naccache, and Mehdi Tibouchi. "Public key compression and modulus switching for fully homomorphic encryption over the integers." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2012.
10. Gentry, C., Halevi, S., & Smart, N. P. (2012). "Fully homomorphic encryption with polylog overhead". In proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, pp. 465-482.
11. Gentry, C., Sahai, A., & Waters, B. (2013). "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based." In proceedings of Annual Cryptology Conference. Springer, Berlin, Heidelberg, pp. 75-92.
12. Brakerski, Z., & Vaikuntanathan, V. (2014). "Efficient fully homomorphic encryption from LWE". SIAM Journal on Computing, vol.43, no.2, pp. 831-871.
13. Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). "(Leveled)" Fully homomorphic encryption without bootstrapping". ACM Transactions on Computation Theory (TOCT), vol.6, no.3, pp. 1-36.
14. Smart, N. P., & Vercauteren, F. (2014). "Fully homomorphic SIMD operations." Designs, codes and cryptography, vol.71, no.1, pp. 57-81.
15. Khalil, I. M., Khreishah, A., & Azeem, M. (2014). "Cloud computing security: A survey. Computers, vol.3, no.1, pp. 1-35.
16. Alperin-Sheriff, J., & Peikert, C. (2014). "Faster bootstrapping with polynomial error". In proceedings of Annual Cryptology Conference. Springer, Berlin, Heidelberg, pp. 297-314
17. Brakerski, Z., & Vaikuntanathan, V. (2014). "Lattice-based FHE as secure as PKE." In Proceedings of the 5th conference on Innovations in theoretical computer science, Association for Computing Machinery, New York, NY, United States, pp. 1-12.
18. Cheon, Jung Hee, and Damien Stehlé. "Fully homomorphic encryption over the integers revisited." In Proceeding of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2015.
19. Ducas, L., & Micciancio, D. (2015). "FHEW: bootstrapping homomorphic encryption in less than a second". In proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, pp. 617-640.
20. Aslett, L. J., Esperança, P. M., & Holmes, C. C. (2015). "A review of homomorphic encryption and software tools for encrypted statistical machine learning". Researchgate: stat.1508.06574, pp. 20-26.
21. Chillotti, I., Gama, N., Georgieva, M., & Izabachene, M. (2016). "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds." In proceedings with international conference on the theory and application of cryptology and information security. Springer, Berlin, Heidelberg, pp. 3-33.
22. Iqbal, S., Kiah, M. L. M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R. (2016). "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service." Journal of Network and Computer Applications, vol.74 no.1, pp. 98-120.
23. Hariss, Khalil, Hassan Noura, and Abed Ellatif Samhat. "Fully Enhanced Homomorphic Encryption algorithm of MORE approach for real world applications." Journal of Information Security and Applications vol. 34 (2017): pp.233-242.
24. Martins, P., Sousa, L., & Mariano, A. (2017). "A survey on fully homomorphic encryption: An engineering perspective." ACM Computing Surveys (CSUR), vol. 50 no.6, pp. 1-33.
25. Ma, Chunguang, Juyan Li, and Weiping Ouyang. "Lattice-based identity-based homomorphic conditional proxy re-encryption for secure big data computing in cloud environment." International Journal of Foundations of Computer Science vol. 28.no.06 (2017): pp.645-660.
26. Kim, Jinsu, Sungwook Kim, and Jae Hong Seo. "A new scale-invariant homomorphic encryption scheme." Information Sciences vol.422 (2018): pp.177-187.
27. Potey, Manish M., C. A. Dhote, and Deepak H. Sharma. "Minimizing Ciphertext in Homomorphic Encryption Scheme for Cloud Data." In proceedings of Progress in Advanced Computing and Intelligent

- Engineering. Springer, Singapore, 2018. pp.583-591.
28. Shen, Wenting, et al. "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage." *IEEE Transactions on Information Forensics and Security* vol. 14, no. 2 (2018): pp.331-346.
 29. Mahmood, Z. H., & Ibrahim, M. K. (2018)." New fully homomorphic encryption scheme based on multistage partial homomorphic encryption applied in cloud computing". In proceedings of 2018 1st Annual International Conference on Information and Sciences (AiCIS). IEEE, pp. 182-186.
 30. Kumar, P. R., Raj, P. H., & Jelciana, P. (2018)." Exploring data security issues and solutions in cloud computing." *Procedia Computer Science*, vol.125 no.1, pp. 691-697.
 31. Cheon, J. H., and Han, K. "Bootstrapping for approximate homomorphic encryption." In proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Cham, 2018.
 32. Angel, S., Chen, H., Laine, K., & Setty, S. (2018)." PIR with compressed queries and amortized query processing." In proceedings of 2018 IEEE symposium on security and privacy (SP). IEEE, pp. 962-979.
 33. El Makkaoui, K., Beni-Hssane, A., & Ezzati, A. (2019)." Speedy Cloud-RSA homomorphic scheme for preserving data confidentiality in cloud computing." *Journal of Ambient Intelligence and Humanized Computing*, vol.10 no.12, pp. 4629-4640.
 34. Dyer, James, Martin Dyer, and Jie Xu. "Practical homomorphic encryption over the integers for secure computation in the cloud." *International Journal of Information Security* vol.18.no.5 (2019): pp.549-579.
 35. Agarkar, Aarti Amod, and Himanshu Agrawal. "Lightweight R-LWE-based privacy preservation scheme for smart grid network." *International Journal of Information and Computer Security* vol.11.no.3 (2019): pp.233-254.
 36. Kiss, T., Kacsuk, P., Kovács, J., Rakoczi, B., Hajnal, Á., Farkas, A, Terstyanszky, G. (2019)." Micado—microservice-based cloud application-level dynamic orchestrator." *Future Generation Computer Systems*, vol.94 no.3, pp. 937-946.
 37. 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*, vol.48, pp. 102362.
 38. Gentry, C., & Halevi, S. (2019, December)." Compressible FHE with applications to PIR." In proceedings of Theory of Cryptography Conference. Springer, Cham, pp. 438-464
 39. Zhou, S., Yu, Z., Nasr, E. S. A., Mahmoud, H. A., Awwad, E. M., & Wu, N. (2020)." Homomorphic encryption of supervisory control systems using automata." *IEEE Access*, vol.8 no.3, pp. 147185-147198.
 40. Elmahdi, E., Yoo, S. M., & Sharshembiev, K. (2020)." Secure and reliable data forwarding using homomorphic encryption against blackhole attacks in mobile ad hoc networks." *Journal of Information Security and Applications*, vol.51, no.3, pp. 102425.
 41. Arita, S., & Handa, S. (2020). Fully Homomorphic Encryption Scheme Based on Decomposition Ring. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.103. no.1, pp.195-211.
 42. Al Badawi, A., Veeravalli, B., Lin, J., Xiao, N., Kazuaki, M., & Mi, A. K. M. (2020)." Multi-GPU design and performance evaluation of homomorphic encryption on GPU clusters." *IEEE Transactions on Parallel and Distributed Systems*, vol.32, no.2, pp. 379-391.
 43. Boura, C., Gama, N., Georgieva, M., & Jetchev, D. (2020)." Chimera: Combining ring-LWE-based fully homomorphic encryption schemes." *Journal of Mathematical Cryptology*, vol. 14, no.1, pp. 316-338
 44. Alharbi, A., Zamzami, H., & Samkri, E. (2020)." Survey on Homomorphic Encryption and Address of New Trend." *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no.7, pp. 618-626.
 45. Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2020)." TFHE: fast fully homomorphic encryption over the torus." *Journal of Cryptology*, vol. 33, no.1, pp. 34-91.
 46. Alabdulatif, A., Khalil, I., & Yi, X. (2020)." Towards secure big data analytics for cloud-enabled applications with fully homomorphic encryption." *Journal of Parallel and Distributed Computing*, vol.137 no.2, pp. 192-204.
 47. Zuber, M., Carpov, S., & Sirdey, R. (2020). Towards real-time hidden speaker recognition by means of fully homomorphic encryption." In Proceedings of the International Conference on Information and Communications Security. Springer, Cham, pp. 403-421
 48. Blatt, M., Gusev, A., Polyakov, Y., & Goldwasser, S. (2020)." Secure large-scale genome-wide association studies using homomorphic encryption. *Proceedings of the National Academy of Sciences*, vol.117 no.21, pp. 11608-11613
 49. Kalyani, P., Masooda, M., & Namrata, P. (2021). Preserving Privacy of Data in Distributed Systems Using Homomorphic Encryption." In proceedings with Progress in Advanced Computing and Intelligent Engineering. Springer, pp. 306-313.
 50. Chillotti, I., Ligier, D., Orfila, J. B., & Tap, S. (2021)." Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for TFHE." In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, pp. 670-699.

51. Chen, Y., Dong, S., Li, T., Wang, Y., & Zhou, H. (2021). "Dynamic multi-key FHE in asymmetric key setting from LWE." In Proceedings of Transactions on Information Forensics and Security, IEEE, pp. 5239-5249.
52. Samardzic, N., Feldmann, A., Krastev, et al. (2021). F1: A fast and programmable accelerator for fully homomorphic encryption." In proceedings of MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 238-252.
53. Jung, W., Kim, S., Ahn, J. H., Cheon, et al. (2021). "Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs." In proceedings of IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 114-148.
54. Mittal, Sonam, and K. R. Ramkumar. (2021) "Research perspectives on fully homomorphic encryption models for cloud sector." Journal of Computer Security, vol. 29, no. 2, pp. 135-160.
55. Yousuf, H., Lahzi, M., Salloum, S. A., & Shaalan, K. (2021). Systematic Review on Fully Homomorphic Encryption Scheme and Its Application. In proceedings of Recent Advances in Intelligent Systems and Smart Applications, pp. 537-551.
56. Gürsoy, G., Chielle, E., Brannon, C. M., Maniatakos, M., & Gerstein, M. (2022). "Privacy-preserving genotype imputation with fully homomorphic encryption. Cell Systems, vol. 13, no.2, pp. 173-182.
57. Mittal, Sonam, Shagun Sharma, and K. R. Ramkumar. "A Matrix-Based Homomorphic Encryption for Preserving Privacy in Clouds." ECS Transactions vol. 107, no.1 (2022): pp.5441.
58. Rupa, C., Greeshmanth, & Shah, M. A. "Novel secure data protection scheme using Martino homomorphic encryption". Journal of Cloud Computing, vol. 12, no. 1 (2023): pp.47.
59. Mittal, S., & Ramkumar, K. R. "Comparative evaluation of fully homomorphic encryption algorithms in cloud environment". International Journal of Electronic Security and Digital Forensics, vol. 15, no. 4 (2023): pp. 333-347
60. Boudgoust, K., & Scholl, P. (2023, December). Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 371-404). Singapore: Springer Nature Singapore.
61. Madni, H. A., Umer, R. M., & Foresti, G. L. (2023). Swarm-FHE: fully homomorphic encryption-based swarm learning for malicious clients. International journal of neural systems, 33(08), 2350033.
62. Mahato, G. K., & Chakraborty, S. K. (2023). A comparative review on homomorphic encryption for cloud security. IETE Journal of Research, 69(8), 5124-5133.
63. Petrean, D. E., & Potolea, R. (2024). Homomorphic encrypted Yara rules evaluation. Journal of Information Security and Applications, 82, 103738.
64. Manulis, M., & Nguyen, J. (2024, May). Fully homomorphic encryption beyond IND-CCA1 security: Integrity through verifiability. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 63-93). Cham: Springer Nature Switzerland.
65. Crihan, G., Dumitriu, L., & Crăciun, M. V. (2024). Preliminary Experiments of a Real-World Authentication Mechanism Based on Facial Recognition and Fully Homomorphic Encryption. Applied Sciences, 14(2), 718.
66. Wang, X., & Li, P. (2024). A Secure Face Verification Scheme Based on Fully Homomorphic Encryption with Anonymity. Information, 15(3), 129.
67. Valera-Rodríguez, F. J., Manzanares-López, P., & Cano, M. D. (2024). Empirical Study of Fully Homomorphic Encryption Using Microsoft SEAL. Applied Sciences, 14(10), 4047.
68. Rieyan, S. A., News, M. R. K., Rahman, A. M., Khan, S. A., Zaarif, S. T. J., Alam, M. G. R., ... & Fortino, G. (2024). An advanced data fabric architecture leveraging homomorphic encryption and federated learning. Information Fusion, 102, 102004.
69. Cheon, S., Lee, Y., Kim, D., Lee, J. M., Jung, S., Kim, T., ... & Kim, H. (2024). {DaCapo}: Automatic Bootstrapping Management for Efficient Fully Homomorphic Encryption. In 33rd USENIX Security Symposium (USENIX Security 24) (pp. 6993-7010).
70. K. D., K. R. Ramkumar and S. Mittal, (2025) "A Comprehensive Survey of Fully Homomorphic Encryption to Mitigate Server-Side Attacks in Cloud Communications," Proceedings of the Innovations and Emerging Technologies in AI & Communication Systems (IETACS), pp. 8-14.