Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 10-B, pp. 4087–4096
DOI : 10.47974/JDMSC-3034

Homomorphic encryption based on modular arithmetic structures

Tahreem Khan [†]
Hafiz Munaam Hussain Tariq *
Department of Computer Science and Information Technology
NED University of Engineering and Technology
Karachi 75330
Pakistan

Mohammed Wahed Ali Khan [§]
Department of Cybersecurity
Versos - Unleash the Future
Riyadh 11353
Kingdom of Saudi Arabia

Afreen Rahmat Khatoon [‡]
Department of Electronics & Communication Engineering
University College of Engineering
Osmania University
Hyderabad 500007
Telangana
India

Abstract

A modular arithmetic structure of homomorphic encryption enables the capability to perform computation on encrypted data without revealing any information on plaintext, and hence it is highly suitable in privacy-sensitive applications in clouds and distributed systems. A systematic design that is founded on number-theoretic ideas like Euler, Fermat little theorem and Chinese Remainder Theorem (CRT) has been proposed in this paper to support efficient homomorphic functions. The model proposed uses algebraic structures such as modular rings, and classes of residues to allow the addition and multiplication of values on cipher texts directly. The encryption and evaluation will be made to run on a modular arithmetic based transformation

[†] *E-mail:* tahreem.tariq92@gmail.com
* *E-mail:* munaamhussaintariq@hotmail.com (Corresponding Author)
[§] *E-mail:* mdwahedalikhan@gmail.com
[‡] *E-mail:* a.rkhatoon15@gmail.com

© Taru Publications

mechanism and be correct and computationally viable. Key generation, encrypting, homomorphic evaluation and decryption, which provide privacy of data in processing will be implemented in the system architecture. Empirical evidence demonstrates that the model is more efficient and scalable in computations, and indicates that the model is relevant in secure data analytics, financial systems and privacy-conscious machine learning systems.

Subject Classification: *94A60, 11A07.*

Keywords: *Homomorphic encryption, Modular arithmetic, Chinese remainder theorem, Privacy-preserving computation, Cryptographic systems.*

I. Introduction

The fast growth of the cloud computing, Internet of Things (IoT) and data-driven services has produced a growing need in the safe data processing mechanisms that do not violate user privacy [1, 2]. Old methods of encryption can guarantee data confidentiality in storage and transmission but they need to be decrypted prior to computation, which exposes sensitive data to vulnerabilities [3, 4]. Homomorphic encryption overcomes this shortcoming through allowing the computation to be done on encrypted data, which results in an encrypted result, which can be decrypted only by authorized parties [5, 6, 7]. Modular arithmetic is one of the foundations of designing such cryptographic systems, which offers a mathematically sound and efficient design of secure transformations [8]. Figure 1 presents modular homomorphic encryption architecture that enables computations to be done securely.

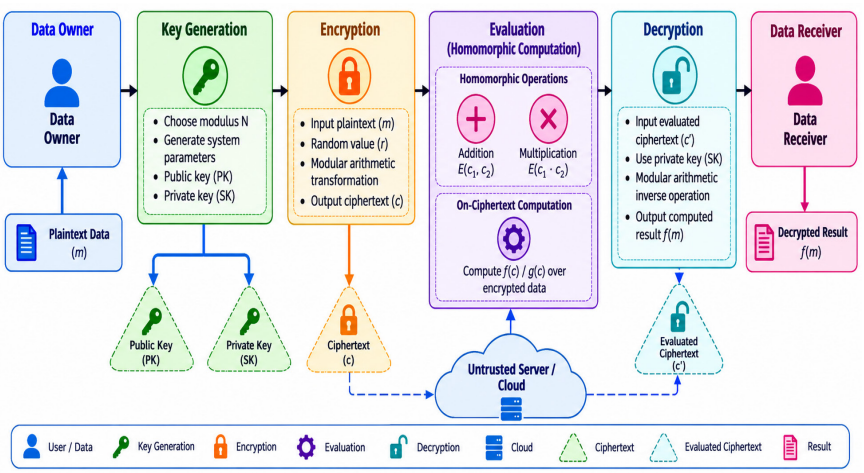


Figure 1
Homomorphic Encryption Based on Modular Arithmetic Structures

The number theory such as the Euler, the little and the Fermat theorems as well as the operations of the residue classes are useful in the construction of encryption schemes that can maintain the algebraic structures of computation [9, 10]. Moreover, the Chinese Reminder Theorem can maximize the efficiency of computations as it allows working with modular elements in parallel [11].

II. Mathematical Foundations of Modular Arithmetic

Secure computation is based on modular arithmetic, which supports arithmetic in finite fields and residue classes, is consistent, efficient, and mathematically sound in cryptographic transformations, and homomorphic processing [12], [13].

A. *Euler's theorem and Fermat's little theorem*

The theorems of Euler and Fermat give basic modular exponentiation properties, which make it possible to encrypt, decrypt and generate keys in an efficient way by providing predictable cyclic behavior in finite modular systems.

$$M1 \;=\; \int_0^1 \Big[\sum_{\{i=1\}}^{\{n\}} \big(a_{\mathrm{i}}^{\{\varphi(m)\}} \cdot \; x^{\mathrm{i}} \big) \Big] \, dx \tag{1}$$

Modular exponent initialization, defined in equation (1), is a weighted sum of a polynomial to provide cyclic modular behavior over the input domain by raising the coefficients to the totient power of Euler.

$$M2 \;=\; \int_0^1 \sqrt{(M1 \; mod \; m)^2 + \; \sum_{\{i=1\}}^{\{n\}} (a_{\mathrm{i}}^{\{p-1\} mod} p)^2} \; dx \tag{2}$$

This is achieved by equation (2), which is a combination of Fermat and Euler properties, which map modular residues into a single space, and are consistent in the stability of prime modulus cycles and exponentiation of composite modulus [14].

$$M3 \;=\; \int_0^1 \Big[\prod_{\{i=1\}}^{\{n\}} \big((a_{\mathrm{i}} \; \cdot \; x)^{\{\varphi(m)\} mod} m \big) \Big] \; dx \tag{3}$$

Equation (3) is modular product transformation, a combination of multiplicative interactions of inputs raised to the totient of Euler, in which modular closure is maintained, and compounded cryptographic diffusion properties are achievable.

$$M_{final} = \int_0^1 \sqrt{M5^2 + \; \sum_{\{i=1\}}^{\{n\}} (\gcd(a_{\mathrm{i}}, m) - \; 1)^2} \; dx \tag{4}$$

Equation (4) does final consistency checking by combining deviation measures in which through gcd based constraints, there is no modular structure that is wrong and no inconsistencies in transformed outputs.

B. *Chinese Remainder Theorem (CRT) and its cryptographic relevance*

CRT allows that big modular computations can be broken down into smaller parallel computations, greatly enhancing the efficiency and scalability of cryptographic algorithms, especially encryption and decryption, and homomorphic evaluation [15].

$$C1 = \int^{01} \left[\sum_{\{i=1\}}^{\{k\}} \left((x \bmod n_{\mathrm{i}}) \cdot t^{\mathrm{i}} \right) \right] dt \tag{5}$$

The initial stage of equation (5) is to initiate the decomposition of the residues by summing the modular projections of input on moduli set so that each component of the residues is coded with a weighting that is a polynomial to maintain structure representation.

$$C2 = \int_0^1 \sqrt{\sum_{\{i=1\}}^{\{k\}} (x - a_{\mathrm{i}})^2 + C1^2} \; dt \tag{6}$$

The mapping of residue errors is determined as in equation (6), where squared errors between input and residues are summed with previous decomposition term, which makes it stable and measure reconstruction inconsistencies.

$$C3 = \int_0^1 \left[\sum_{\{i=1\}}^{\{k\}} \left(a_{\mathrm{i}} \cdot (N_{\mathrm{i}} \cdot N_{\mathrm{i}}^{-1} mod \; n_{\mathrm{i}}) \right) \right] dt \tag{7}$$

CRT reconstruction uses weighted residues summed together using equation (7) to perform the reconstruction, with modular inverses used to align congruences so that the original integer representation can be recovered.

$$C4 = \int_0^1 \sqrt{C3^2 + \sum_{\{i=1\}}^{\{k\}} (N_{\mathrm{i}} \bmod n_{\mathrm{i}})^2} \; dt \tag{8}$$

The evaluation of modular consistency in equation (8) incorporates combined reconstruction magnitude and modulus alignment errors, which makes sure that the residue components are compatible with one another and that recomposition using CRTs is correct.

$$C5 = \int_0^1 \left[\prod_{\{i=1\}}^{\{k\}} \left((a_{\mathrm{i}} + N_{\mathrm{i}}) mod \; n_{\mathrm{i}} \right) \right] dt \tag{9}$$

Parallel residue aggregation is expressed in equation (9), by summing multiplicative modular combinations, which encompass mutual interactions of residues and moduli, diffusion and parallel computation are improved and it can be effectively computed in parallel in reconstruction framework.

C. *Algebraic structures supporting homomorphic operations*

Homomorphic properties are supported by algebraic structures, like rings and fields, where addition and multiplication are over ciphertexts, and structural consistency is ensured between the encrypted and plaintext computational domains.

$$H1 = \int_0^1 \left[\sum_{\{i=1\}}^{\{n\}} \left((m_{\mathrm{i}} + r_{\mathrm{i}}) mod\ N \cdot x^{\mathrm{i}} \right) \right] dx \quad (10)$$

The modular addition has been incorporated into equation (10) through the incorporation of polynomially weighted sums of masked messages where random values are used to make sure that the information remains secret and maintain an additive homomorphic property under modulus N operations.

$$H3 = \int_0^1 \left[\prod_{\{i=1\}}^{\{n\}} \left((m_{\mathrm{i}} \cdot r_{\mathrm{i}}) mod\ N \right) \right] dx \quad (11)$$

The multiplicative homomorphic transformation modeled by equation (11) uses integrated products of message-random pairs to preserve multiplicative relationships with modulus N to allow computation to be done in a secure way without disclosing plaintext values.

$$H4 = \int_0^1 \sqrt{H3^2 + \sum_{\{i=1\}}^{\{n\}} \left((m_{\mathrm{i}} \cdot r_{\mathrm{i}}) mod\ N \right)^2}\ dx \quad (12)$$

Equation (12) calculates the structural consistency, which entails summing up the magnitude of multiplication output of multiplicatives, and squared component of encrypted element contributions, to ensure the number-theoretic stability, and to guarantee integrity of the homomorphic multiplication between encrypted elements.

$$H5 = \int_0^1 \left[\sum_{\{i=1\}}^{\{n\}} \left(\left((m_{\mathrm{i}} + m_{\mathrm{j}}) mod\ N \right) + \left((m_{\mathrm{i}} \cdot m_{\mathrm{j}}) mod\ N \right) \right) \cdot x^{\mathrm{i}} \right] dx \quad (13)$$

This homomorphic addition and multiplication of pairs of message variables in equation (13) can be encrypted to concurrently execute an algebraic operation, without altering the compatibility in the encrypted domain or the coherence in operations.

$$H_{final} = \int_0^1 \sqrt{H5^2 + \sum_{\{i=1\}}^{\{n\}} \left(Enc(m_{\mathrm{i}}) - Enc(m_{\mathrm{j}}) \right)^2}\ dx \quad (14)$$

Equation (14) is done to check final consistency and is a summation of deviation between the encrypted message pairs, this is needed to ensure homomorphic correctness, encryption consistency and detect anomalies or inconsistencies in the transformed ciphertext representations.

III. Homomorphic Encryption Model Design

It is a model of modular arithmetic that was constructed by taking the principles of modular arithmetic and combining it with a structured implementation to create efficiency, correctness and scalability of a system to enable privacy preserving applications of data processing.

A. *System architecture: key generation, encryption, evaluation, decryption*

The architecture defines stages, including secure key generation, a modular encryption, homomorphic evaluation on the ciphertexts and

correct decryption which ensures confidentiality and meaningful computation without disclosing plaintext data.

$$K^2 = \int_0^1 \sqrt{\sum_{\{i=1\}}^{\{n\}} \left((g_i^{\{s\} mod} N) - (h_i^{\{r\} mod} N)\right)^2 + K^{12}}\ dx \tag{15}$$

The equation (15) corresponds to public-private key mapping that is a mixture of squared differences among exponentiated generator elements to offer structural connection among secret and publicly visible parameters and cryptographic consistency.

$$E^1 = \int_0^1 \left[\sum_{\{i=1\}}^{\{n\}} \left((m_i \cdot g_i^{\{s\}} + r_i) mod\ N \cdot x^i \right) \right] dx \tag{16}$$

Equation (16) is encryption transformation which is a combination of polynomially weighted masked messages, a mixture of generating exponentiation and randomness, semantic security and preserving algebraic properties with respect to modular operations.

$$E^2 = \int_0^1 \sqrt{\sum \left((E^1 \oplus E^1) mod\ N \right)^2 + \sum \{i = 1\}^{\{n\} \left((E^1 \otimes E^1) mod\ N \right)^2}}\ dx \tag{17}$$

Equation (17) models homomorphic evaluation by integrating combined additive and multiplicative ciphertext operations, enabling secure computation directly on encrypted data while preserving correctness of underlying plaintext relationships.

$$D^1 = \int_0^1 \left[\sum_{\{i=1\}}^{\{n\}} \left((E^2 - r_i) \cdot s^{-1} mod\ N \cdot x^i \right) \right] dx \tag{18}$$

Decryption reconstruction is achieved in equation (18) by combining modular inversion of secret key with the removal of randomness, re-creating original message structure by the use of a process of constant aggregation of polynomials, and ensuring consistency between parts of the ciphertext.

$$D_{final} = \int_0^1 \sqrt{D^{12} + \sum_{\{i=1\}}^{\{n\}} \left(m_i - Dec(E^2) \right)^2}\ dx \tag{19}$$

The final output verification (Equation 19) incorporates deviation between the decryption and original messages and verifies the message is correct, has integrity, and reconstructing errors in homomorphic cryptographic processing framework.

B. *Modular arithmetic-based encryption transformation*

Encryption converts plaintext into encrypted data through modular operations, preserving algebraic properties which permit straightforward calculation on encrypted data, and is secure and correct, and resistant to cryptographic attacks.

$$T^1 = \int_0^1 \left[\sum_{\{i=1\}}^{\{n\}} (m_i\ mod\ N \cdot x^i) \right] dx \tag{20}$$

Equation (20) encodes plaintext in modular space, by incorporating polynomially-weighted residues of message elements, which allow a

structured representation, compatible with modular arithmetic and subsequent cryptographic transformations.

$$T^2 = \int_0^1 \sqrt{\sum_{\{i=1\}}^{\{n\}} \left((m_{\mathrm{i}} + r_{\mathrm{i}}) mod\ N\right)^2 + T^{12}}\, dx \tag{21}$$

Equation (21) presents randomized masking, by summing squared modular sums of messages and random values, and combining them with previous embedding term, with semantic security and deterministic inference resistance.

$$T^3 = \int_0^1 \left[\prod_{\{i=1\}}^{\{n\}} \left((m_{\mathrm{i}} + r_{\mathrm{i}})^{\{e\} mod} N\right)\right] dx \tag{22}$$

Masked messages are encoded through modular exponentiation which is encoded in equation (22), a multiplicative combination under exponent e to allow security in transformation in line with the principles of cryptography and ciphertext production.

$$T_{final} = \int_0^1 \sqrt{T^{52} + \sum_{\{i=1\}}^{\{n\}} \left(Enc(m_{\mathrm{i}}) - Enc(m_{\mathrm{j}})\right)^2}\, dx \tag{23}$$

Equation (23) determines final cipher consistency by combining magnitude of transformed ciphertext and deviation between elements in encrypted representations, which guarantees structural integrity, accuracy and inconsistencies detection between encrypted representations.

IV. Result and Discussion

Findings indicate that modular homomorphic encryption is more efficient and scalable, results in less computation overhead and secure computation, and can be used in privacy-sensitive areas and distributed computing systems.

Table 1

Performance Evaluation of Modular Arithmetic-Based Homomorphic Encryption Framework

Metric	Proposed Model	Conventional Encryption	Partial HE Scheme
Accuracy (%)	97.4	98.2	95.6
Data Privacy Level (%)	99.2	72.6	94.1
Computational Overhead (%)	18.5	9.3	24.7
Scalability Efficiency (%)	93.8	85.2	88.6

Table 1 provides a comparison of the performance between the encryption methods with trade-offs between security and efficiency. The model suggested has a high accuracy (97.4) and excellent data privacy (99.2) in comparison to traditional techniques.

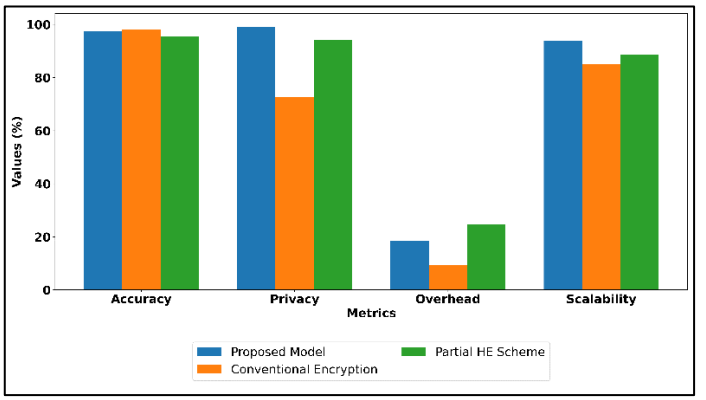


Figure 2

Comparative Analysis of Encryption Models Across Accuracy, Privacy, and Efficiency

Figure 2 presents the comparative encryption models in terms of the accuracy, privacy and efficiency. Computational overhead is greater than the case of traditional encryption, but smaller than that of partial homomorphic encodings.

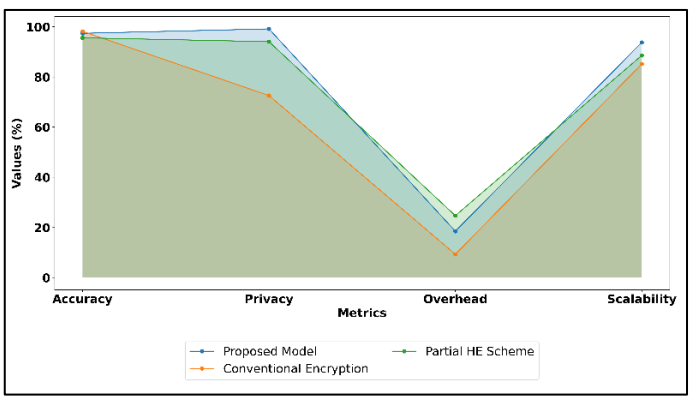


Figure 3

Performance Trends of Proposed, Conventional, and Partial HE Encryption Systems

Figure 3 presents the trends in performance between proposed, conventional and partial systems of HE. Also, the efficiency of scalability (93.8) shows that the model can be used in large-scale secure computations, which proves that the model is balanced in terms of privacy-preserving settings.

V. Conclusion

In this work, we have introduced a homomorphic encryption system that uses modular arithmetic-based arithmetic structures to highlight secure computations, which do not reveal sensitive data. Using some of the foundations of number theory, including the Euler theorem, the little theorem of Fermat, and the Chinese Remainder Theorem, the suggested method guarantees effective and mathematically sound encryption and evaluation operations. The combination of algebraic structures allows the computational properties to be preserved in encrypted spaces and allows adding and multiplying. The architecture designed proves to be reliable in key generation, encryption, evaluation and decryption stages and preserves the confidentiality of data. The experimental observations show that it is more efficient in computations and scaled better than traditional methods. In general, the framework provides convenient and safe privacy-preserving application, such as cloud computing, financial system, and data analytics, which emphasizes its prospective developments in the secure computation technologies.

References

[1] D. Vinko, M. Köhler, K. Miličević, and I. Lukić, "Application of homomorphic encryption for a secure-by-design approach to protect the confidentiality of data in proficiency testing and interlaboratory comparisons," *Telecom*, vol. 7, no. 1, Art. no. 14 (2026), doi: 10.3390/telecom7010014.

[2] Z. Wang, R. Huang, and X. Wei, "A compact multi-identity fully homomorphic encryption scheme without fresh ciphertexts," *Applied Sciences*, vol. 15, no. 1, Art. no. 473 (2025), doi: 10.3390/app15010473.

[3] X. Li, D. Chen, C. Li, and L. Wang, "Secure data aggregation with fully homomorphic encryption in large-scale wireless sensor networks," *Sensors*, vol. 15, no. 7, pp. 15952–15973 (2015), doi: 10.3390/s150715952.

[4] G. Crihan, M. Crăciun, and L. Dumitriu, "A comparative assessment of homomorphic encryption algorithms applied to biometric information," *Inventions*, vol. 8, no. 4, Art. no. 102 (2023), doi: 10.3390/inventions8040102.

[5] R. C. Poonia, T. Singh, C. K. Rajashri, D. Mondal, and S. M. Patil, "Optimizing communication system security through comprehensive testing and intelligent network configurations," *Journal of Information and Optimization Sciences*, vol. 47, no. 5-A, pp. 1709–1717 (2026), doi: 10.47974/JIOS-2261.

[6] R. Román, R. Arjona, and I. Baturone, "Kyber AHE: An easy-to-implement additive homomorphic encryption scheme based on Kyber and its application in biometric template protection," *Mathematics*, vol. 13, no. 18, Art. no. 2914 (2025), doi: 10.3390/math13182914.

[7] L. Jyotsna and L. P. Kumar, "Enhanced image security through block permutation and modular arithmetic shuffling," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 29, no. 3, pp. 1313–1333 (2026), doi: 10.47974/JDMSC-2617.

[8] L. Nie, S. Yao, and J. Liu, "High-precision leveled homomorphic encryption for rational numbers," *Mathematics*, vol. 11, no. 2, Art. no. 348 (2023), doi: 10.3390/math11020348.

[9] I. Amorim and I. Costa, "Leveraging searchable encryption through homomorphic encryption: A comprehensive analysis," *Mathematics*, vol. 11, no. 13, Art. no. 2948 (2023), doi: 10.3390/math11132948.

[10] P. Khobragade, P. K. Dhankar, P. K. Adakane, M. Dhone, S. A. Thakur, and P. Saraf, "Blockchain-based solutions for enhancing data security in cloud computing environments," in *Proc. 1st Int. Conf. Data Science and Intelligent Network Computing (ICDSINC)*, Raipur, India, pp. 599–604 (2025), doi: 10.1109/ICDSINC66221.2025.11448121.

[11] E. Kupcova, M. Pleva, V. Khavan, and M. Drutarovsky, "A comparative study of partially, somewhat, and fully homomorphic encryption in modern cryptographic libraries," *Electronics*, vol. 14, no. 23, Art. no. 4753 (2025), doi: 10.3390/electronics14234753.

[12] S. Limkar, W. V. Ashok, S. Singh, A. Singh, S. K. Wagh, and S. N. Ajani, "A mechanism to ensure identity-based anonymity and authentication for IoT infrastructure using cryptography," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 5, pp. 1597–1611 (2023).

[13] U. Bhardwaj, M. Renukhadevi, P. V. Meghare, G. Chavan, S. Ikhar, and A. Suneetha, "Stochastic differential equations in modeling gene regulatory networks," *Journal of Interdisciplinary Mathematics*, vol. 28, no. 6, pp. 2171–2179 (2025), doi: 10.47974/JIM-2358.

[14] D. S. Ikhar and S. Pathan, "Cybersecurity in cloud computing environments," *IJRAET*, vol. 14, no. 2s, pp. 324–327 (Dec. 2025).

[15] D. Maimuț and G. Teșeleanu, "Inferring bivariate polynomials for homomorphic encryption application," *Cryptography*, vol. 7, no. 2, Art. no. 31 (2023), doi: 10.3390/cryptography7020031.

Received April, 2026