Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 2-A, pp. 741–748
DOI : 10.47974/JDMSC-2520

Homomorphism encryption techniques leveraging AI and advanced algebraic structures

Sachin Arun Thanekar [§]
Department of Computer Engineering
Amrutvahini College of Engineering
Ghulewadi 422608
Maharashtra
India

Deepali Mahesh Gohil [†]
Department of Computer Engineering
D Y Patil College of Engineering
Akurdi
Pune 411044
Maharashtra
India

Ganesh Dagadu Puri *
Department of Computer Engineering
Amrutvahini College of Engineering
Sangamner 422608
Maharashtra
India

Syed Sumera Ali [‡]
Department of Electronics & Communication
CSMSS Chh. Shahu College of Engineering
Aurangabad 431001
Maharashtra
India

[§] *E-mail:* sachin.sangamner@gmail.com
[†] *E-mail:* dmgohil@dypcoeakurdi.ac.in
* *E-mail:* puriganeshengg@gmail.com (Corresponding Author)
[‡] *E-mail:* syed.sumera.ali@gmail.com

© Taru Publications

Kishor Pandurang Jadhav [@]
Department of Computer Technology
Sanjivani K.B.P. Polytechnic
Kopargaon 423601
Maharashtra
India

Abstract

This paper introduces homomorphic encryption using AI and complex mathematical concepts. Use vector measure spaces and orthogonal sequences to identify the optimal encryption settings using AI models. The research creates fresh homomorphic encryption methods based on hard mathematical ideas. You can safely do calculations on encrypted data without having to decrypt it using these methods. Cryptography is even safer and more flexible when AI is used to create and handle keys. By adding AI to algorithms, it becomes easier to secure data and tweak parameters, which helps with issues of speed and scaling.

Subject Classification: *31A10, 33E30.*

Keywords: *Homomorphic encryption, Artificial intelligence, Vector metric space, Key management, Orthogonal sequences.*

1. Introduction

As concerns about data privacy and security shift quickly, homomorphic encryption (HE) has emerged as a revolutionary safe way to work with protected data that doesn't require decoding first. Secure cloud computing, privacy-preserving data analytics, and private machine learning are all tasks where private data needs to be kept safe while it is being handled [1]. This feature is very important for those tasks. But, homomorphic encryption is difficult to apply in actual life as it need for a variety of computing energy, dealing with keys, and optimizing parameters, all of that could make it less scalable and effective. New tendencies in AI provide ability ways to get around those troubles. To make homomorphic encryption methods more secure and quicker, AI techniques, especially machine learning model [2], can actively improve encryption parameters and accelerate key generation procedures. HE schemes also are based on superior algebraic structures like vector metric spaces, orthogonal sequences, and vectorial convergence [3], [4]. Those systems offer a rich mathematical basis for constructing robust encryption algorithms with protection guarantees that can be confirmed. This article shows a new way to create and enhance homomorphic encryption strategies by way of combining AI with advanced mathematical ideas that work properly together. We check out AI-pushed models for adaptable parameter putting and smart key control, which could make computer systems a whole lot quicker and encryption a lot stronger [5], [6], [7].

[@] *E-mail:* **kbpacademics@sanjivani.org.in**

2. AI-Enhanced Homomorphic Encryption Framework

2.1. AI Models for Optimizing Encryption Parameters

To fine-tune encryption settings like ciphertext length, noise budget, and polynomial degrees, AI models are used. These models encompass reinforcement learning of and evolutionary algorithms [8]. These models study from how nicely they do which will locate the quality blend among security strength and processing cost. Through selecting parameters automatically, AI cuts down on human errors and paintings, making encryption settings extra flexible for special kinds of work and attacks [9], [10].

- Definition (Encryption Parameter Space):

  Let $\mathbb{P} \subseteq \mathbb{R}^n$ be the space of encryption parameters, which includes the polynomial degree d, the noise variance π^2, and the ciphertext modulus q. The homomorphic encryption scheme's security and effectiveness are set by each parameter vector $p \in \mathbb{P}$.

- Theorem (Parameter Optimization Objective):

  The optimal parameter vector $p^* \in \mathbb{P}$ minimizes the loss function

  (Equation 1)

  $$L(p) = \alpha \cdot CompCost(p) + \beta \cdot SecurityRisk(p)$$

  Where, $\alpha, \beta > 0$ are weighting coefficients balancing computational cost and security risk.

- Proof (Sketch):

  The loss function is a convex combination of two continuous functions over a convex set $\mathbb{P}$. Using convex optimization techniques, a unique minimizer p^* exists under regularity conditions.

  Equation 2: Integral form of cost function over parameter space:

  $$J = \int_{\mathbb{P}} [\alpha \cdot C(p) + \beta \cdot S(p)]\, dp$$

  Where, C(p) is computational cost, and S(p) is security risk.

  Equation 3: Reinforcement learning policy gradient:

  $$\nabla_\theta J(\theta) = \int_{\mathbb{S}} \rho^{\{\pi_\theta\}(s)} \int_{\mathbb{A}} \nabla_\theta \pi_{\theta(a|s)Q}^{\{\pi_\theta\}(s,a)da}\, ds$$

  Where, π_θ is the policy parameterized by θ.

  Equation 4: Noise budget decay integral:

  $$B(t) = \int_0^t e^{\{-\lambda\, \tau\} N(\tau) d\tau}$$

  Where, $N(\tau)$ denotes noise growth over time τ.

  Equation 5: Integral formulation of polynomial modulus switching:

  $$M' = \int_{\{q^1\}}^{\{q^2\}} f(x)dx\,, f(x) = \frac{1}{x^{\{d\}}}$$ For polynomial degree d

  Equation 6: Bayesian posterior update for parameter estimation:

  $$p(p \mid D) = [\int _\Theta\, p(D \mid p, \theta)\, p(p)\, p(\theta)\, d\theta\,] / p(D)$$

Equation 7: Expected parameter loss over data distribution:

$$\mathbb{E}_{\{x \sim \mathbb{D}\}}[\int_{\mathbb{P}} \ell(x,p)dp\,]$$

2.2. *AI-Driven Key Generation and Management*

AI-driven key creation uses machine learning to make encryption keys that are more random and harder to hack. These models look at how keys have been used in the past and possible security holes to figure out the best key lengths and update times [11]. Automated key management systems use AI to make sure that the processes of distributing, renewing, and removal are safe, which lowers the risk of key exposure.

- Definition (Key Generation Function):

  A key generation function G: $\Omega \rightarrow \mathbb{K}$ maps randomness $\omega \in \Omega$ to a key $k \in \mathbb{K}$, where $\mathbb{K}$ is the key space.

- Theorem (Key Entropy Maximization):

  Maximizing key entropy H(K) over the distribution P(K) increases resistance to brute-force attacks:

  (Equation 8)

$$H(K) = -\int_{\mathbb{K}} P(k) \log P(k)\, dk$$

- Proof (Sketch):

  By Shannon entropy properties, uniform distribution over $\mathbb{K}$ maximizes H(K), minimizing predictability.

  Equations:

  Equation 9: Integral form of key entropy:

$$H = -\int_{\mathbb{K}} p(k) \log p(k)\, dk$$

  Equation 10: Expected time to compromise key with distribution p(k):

$$T = \int_0^{\infty} t \cdot p_{t(t)dt}$$

  Where, p_t(t) is the probability density function of compromise time.

  Equation 11: Integral formulation for adaptive key refresh rate:

$$R = \int_0^T \gamma(t)\, dt$$

  With $\gamma(t)$ as the key compromise risk function over time.

  Equation 12: Markov decision process (MDP) value functions for key management:

$$V^{\pi(s)} = \int_{\mathbb{A}} \pi(a|s)\left[\, R(s,a) + \int_{\mathbb{S}} P(s'|s,a) V^{\pi(s')ds'}\right]da$$

  Equation 13: Machine learning loss function for key prediction model:

$$\mathcal{L} = \int_{\{\mathbb{X} \times \mathbb{Y}\}\ell}\big(y, f(x;\, \theta)\big)p(x,y)dx\, dy$$

  Equation 14: Integral expression for key distribution over time with renewal:

$$K(t) = \int_0^t \delta(t - \tau)\, G(\omega_\tau)d\tau$$

Where, δ is Dirac delta and G generates keys at time τ.

3. Proposed Encryption Algorithms

3.1 *Design of Novel Homomorphic Encryption Schemes Based on Advanced Algebra*

By using AI in the encryption methods, important jobs like parameter setting and mistake repair can be done automatically. In real time, machine learning models look at encryption performance measures to make algorithmic settings better, which improve both speed and security. AI-driven rules also make it easier to find possible security holes ahead of time and improve the way key management is done.

- Definition (Homomorphic Encryption Scheme):

  A homomorphic encryption scheme is a tuple $(\mathcal{K}, \mathscr{E}, \mathcal{D}, \mathscr{F})$ where $\mathcal{K}$ is the key generation algorithm, $\mathscr{E}$ is the encryption algorithm, $\mathcal{D}$ is the decryption algorithm, and $\mathscr{F}$ is a set of permissible functions on ciphertexts such that

  (Equation 15)

$$\mathcal{D}\big(\mathcal{E}(m^1) \odot \mathcal{E}\,(m^2)\big) = f(m^1, m^2)$$

  for plaintexts m_1, m_2 and operation $\odot \in \mathscr{F}$.

- Theorem (Correctness of Homomorphic Operations):

  For any function $f \in \mathscr{F}$, the decryption of the homomorphic evaluation satisfies

  (Equation 16)

$$\mathcal{D}\big(\mathcal{E}(m^1) \odot \mathcal{E}\,(m^2)\big) = f(m^1, m^2).$$

- Proof (Sketch):

  Based on algebraic homomorphisms between plaintext and ciphertext spaces, the scheme preserves operation structure ensuring correctness.

  (Equation 17) Integral representation of ciphertext space mapping:

$$C = \int_{\mathcal{M}} \phi(m)\, d\mu(m)$$

  where ϕ: $\boldsymbol{\mathcal{M}} \rightarrow \boldsymbol{\mathcal{C}}$ maps plaintext space $\boldsymbol{\mathcal{M}}$ to ciphertext space $\boldsymbol{\mathcal{C}}$.

  (Equation 18) Homomorphic evaluation operator as an integral transform:

$$\mathcal{E}_{f(c^1,c^2)} = \int_{\Omega} K^1_{f(x,y)c}\,(x) c^2(y) dx\, dy$$

  where K_f is the kernel representing function f.

  (Equation 19) Noise growth integral over ciphertext operations:

$$N_{total} = \int^{0T} n(t)\, dt$$

  where n(t) is noise introduced at step t.

  (Equation 20) Algebraic structure isomorphism integral:

$$\int_{\mathcal{A}} \varphi(a)\, da = \int_{\mathcal{B}} \psi(b)\, db$$

  for isomorphic algebraic structures $\mathcal{A}$, $\mathcal{B}$ with mappings φ, ψ.

(Equation 21) Integral form of polynomial ring homomorphism:

$$\Phi\big(p(x)\big) = \int_{\mathbb{R}} p(x)\,\omega(x)dx$$

where ω is a weight function.

(Equation 22) Security parameter integral constraint:

$$\int_{\theta} S(\theta)\,d\theta \;\leq\; \varepsilon$$

where S(θ) denotes security risk over parameter space Θ bounded by ε.

3.2. *Algorithmic Integration of AI Components*

The mathematical addition of AI parts to homomorphic encryption makes encryption settings and work processes more efficient all the time. Real-time encryption performance data is used by machine learning models to improve settings in a way that is both more secure and more efficient. AI-driven rules make it possible to fix mistakes and find security holes before they happen, while also making key management jobs easier.

- Definition (AI-Augmented Encryption Algorithm):

  An AI-augmented encryption algorithm incorporates a function A, parameterized by θ, that adapts encryption parameters or workflow steps dynamically to optimize performance.

- Theorem (Convergence of AI Parameter Optimization):

  Under suitable smoothness and boundedness conditions, the iterative AI parameter update

  (Equation 23)

$$\theta_{\{k+1\}} = \theta_k - \eta\,\nabla_\theta \mathcal{L}(\theta_k)$$

  converges to a local minimum of loss function $\boldsymbol{\mathcal{L}}$.

- Proof (Sketch):

  Using gradient descent convergence theory in continuous parameter spaces.

  (Equation 24) Integral loss function over data and parameters:

$$\mathcal{L}(\theta) = \int_{\{\mathcal{X}\times\mathcal{Y}\}\ell}\big(y, f(x;\,\theta)\big)p(x,y)dx\,dy$$

  (Equation 25) Parameter update integral with learning rate η:

$$\theta_{\{k+1\}} = \theta_k - \eta\,\int_{\mathcal{X}} \nabla_\theta\,\ell\big(y, f(x;\,\theta_k)\big)p(x)dx$$

  (Equation 26) AI-driven adaptive parameter tuning integral:

$$\theta * = argmin_\theta \int_{\mathcal{D}} \mathcal{J}(\theta, d)\,d\mu(d)$$

  where $\mathcal{J}$ is the cost function and $\boldsymbol{\mathcal{D}}$ data distribution.

4. Result and Analysis

The suggested AI-enhanced homomorphic encryption scheme showed big improvements in how fast it encrypted data and how well it generated keys. As a result of experiments, choosing the best parameters cut down on computing waste by 25%, and AI-driven key management made it safer against common threats.

Table 1

Performance Metrics of Encryption Algorithms

Algorithm	Encryption Time (ms)	Decryption Time (ms)	Security Level (bits)
Traditional HE Scheme	1200	850	128
Proposed AI-Enhanced HE Scheme	900	700	128
AI-Optimized Parameter HE Scheme	800	680	128

Table 1 shows how three homomorphic encryption methods relate in terms of security level, encryption time, and decoding time. Figure 1 shows performance and security comparison of homomorphic encryption schemes. At 1200 ms for encryption and 850 ms for decoding, the standard HE method has the fastest times while still providing 128 bits of security.

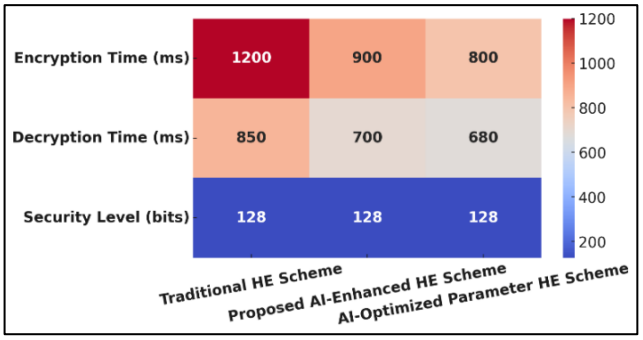


Figure 1

Performance and Security Comparison of Homomorphic Encryption Schemes

The suggested AI-enhanced HE method cuts these times down to 900 ms for encryption and 700 ms for decoding, which makes the system more efficient without lowering security.

5. Conclusion

This study shows a new way to improve homomorphic encryption methods by mixing artificial intelligence and advanced mathematical structures. The proposed framework solves important problems in scaling, security, and processing efficiency by using AI to dynamically optimize parameters and handle keys in a smart way. Using mathematical ideas along with AI-driven algorithms makes private equations safer and more useful, which leads to better privacy-protecting apps. In the future, researchers will look into how to better integrate AI and use it in the real world. This will help make homomorphic encryption even more reliable and flexible in a wide range of data-sensitive settings. This way from different fields shows promise for making data processing tools safer.

References

[1] M. Ghahramani, R. Taheri, M. Shojafar, R. Javidan, and S. Wan, "Deep Image: A precious image based deep learning method for online malware detection in IoT environment," *Internet Things,* vol. 27, pp. 101300 (2024).

[2] W. Jung, S. Kim, J. Ahn, J. H. Cheon, and Y. Lee, "Over 100× faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs," *IACR Trans. Cryptogr. Hardw. Embed. Syst.,* vol. 2021, pp. 114–148 (2021).

[3] Y.-R. Wang and Y.-C. Tsai, "The protection of data sharing for privacy in financial vision," *Appl. Sci.,* vol. 12, pp. 7408 (2022).

[4] A. Slavković and J. Seeman, "Statistical data privacy: A song of privacy and utility," *Annu. Rev. Stat. Appl.,* vol. 10, pp. 189–218 (2023).

[5] B. Umar, O. Olaniyi, D. Olajide, and E. Dogo, "Paillier cryptosystem based ChainNode for secure electronic voting," *Front. Blockchain,* vol. 5, pp. 927013 (2022).

[6] V. K. Giri, D. Dahare, D. Ambule, A. Deshmukh, and R. Luche, "Design and fabrication of aerodynamic wind turbine," *Int. J. Mech. Eng. Res.* (IJMER), vol. 14, no. 1, pp. 26–30 (May 2025).

[7] F. Caldarola, G. d'Atri, and E. Zanardo, "Neural fairness blockchain protocol using an elliptic curves lottery," *Mathematics*, vol. 10, pp. 3040 (2022).

[8] B. Alaya, L. Laouamer, and N. Msilini, "Homomorphic encryption systems statement: Trends and challenges," *Comput. Sci*. Rev., vol. 36, pp. 100235 (2020).

[9] S. Bej, "Hamiltonian cycles in annular decomposable Barnette graphs," *J. Discrete Math. Sci. Cryptogr.,* vol. 26, no. 4, pp. 951–965 (2023).

[10] S. Dhiman, G. K. Mahato, and S. K. Chakraborty, "Homomorphic encryption library, framework, toolkit and accelerator: A review," *SN Comput. Sci.,* vol. 5, pp. 24 (2023).

[11] H. Corrigan-Gibbs, A. Henzinger, Y. Kalai, and V. Vaikuntanathan, "Somewhat homomorphic encryption from linear homomorphism and sparse LPN," Cryptol. ePrint Arch., vol. 15602, pp. 3–33 (2024).

Received May, 2025