APPLICATION OF FULLY HOMOMORPHIC ENCRYPTION IN SECURING AND PROCESSING ELECTRICAL ENERGY CONSUMPTION DATA IN SMART BUILDINGS

Le Hoan, Nguyen Tung Linh*
Electric Power University

ARTICLE INFO

Received: 10/4/2025
Revised: 21/5/2025
Published: 22/5/2025

KEYWORDS

Homomorphic encryption
Data encryption
Security
Privacy
Big data
Digital energy

ABSTRACT

Homomorphic encryption is an emerging cryptographic method that allows data to be processed while still encrypted, offering a promising solution for maintaining privacy in sensitive applications. In this study, we present a practical implementation of the homomorphic encryption aimed at real-world scenarios, with a particular focus on the electricity and energy sectors, domains where data security and confidentiality are critical. We evaluate the performance of the homomorphic encryption by applying it to encrypted energy consumption data and demonstrate that it enables meaningful computations without compromising privacy. Our results show that despite computational overhead, optimizations make homomorphic encryption viable for secure analytics, with a key advantage over AES and RSA: computing directly on encrypted data. This is crucial for smart buildings, where energy usage data must be processed securely. FHE ensures privacy by allowing remote analysis of encrypted electricity data, preventing leaks or misuse. The reliability of FHE makes it a promising solution for securing energy systems and critical infrastructure in privacy-focused smart environments.

ỨNG DỤNG MÃ HÓA ĐỒNG CẤU ĐẦY ĐỦ TRONG VIỆC BẢO VỆ VÀ XỬ LÝ DỮ LIỆU TIÊU THỤ ĐIỆN TRONG CÁC TÒA NHÀ THÔNG MINH

Lê Hoàn, Nguyễn Tùng Linh*
Trường Đại học Điện Lực

THÔNG TIN BÀI BÁO

Ngày nhận bài: 10/4/2025
Ngày hoàn thiện: 21/5/2025
Ngày đăng: 22/5/2025

TỪ KHÓA

Mã hoá đồng cấu
Mã hoá dữ liệu
Bảo mật
Tính riêng tư
Dữ liệu lớn
Năng lượng số

TÓM TẮT

Mã hóa đồng cấu là một phương pháp mã hóa mới nổi cho phép xử lý dữ liệu trong khi vẫn được mã hóa, mang đến giải pháp đầy hứa hẹn để duy trì quyền riêng tư trong các ứng dụng nhạy cảm. Trong nghiên cứu này, chúng tôi trình bày một triển khai thực tế của mã hoá đồng cấu hướng đến các tình huống thực tế, đặc biệt tập trung vào các lĩnh vực điện và năng lượng, các lĩnh vực mà bảo mật và tính bảo mật dữ liệu là rất quan trọng. Chúng tôi đánh giá hiệu suất của mã hoá đồng cấu bằng cách áp dụng nó vào dữ liệu tiêu thụ năng lượng được mã hóa và chứng minh rằng nó cho phép tính toán có ý nghĩa mà không ảnh hưởng đến quyền riêng tư. Kết quả của chúng tôi cho thấy, mặc dù tồn tại chi phí tính toán, các kỹ thuật tối ưu hóa giúp mã hoá đồng cấu trở nên khả thi trong phân tích dữ liệu an toàn, với lợi thế nổi bật so với AES và RSA: cho phép tính toán trực tiếp trên dữ liệu đã mã hoá. Điều này rất quan trọng đối với các tòa nhà thông minh, nơi dữ liệu sử dụng năng lượng phải được xử lý an toàn. FHE đảm bảo quyền riêng tư bằng cách cho phép phân tích từ xa dữ liệu điện được mã hóa, ngăn ngừa rò rỉ hoặc sử dụng sai mục đích. Độ tin cậy của FHE khiến nó trở thành giải pháp đầy hứa hẹn để bảo mật các hệ thống năng lượng và cơ sở hạ tầng quan trọng trong các môi trường thông minh tập trung vào quyền riêng tư.

DOI: https://doi.org/10.34238/tnu-jst.12527

* Corresponding author. *Email: linhnt@epu.edu.vn*

1. Introduction

In the contemporary digital landscape, the proliferation of data across various sectors necessitates robust security measures to protect sensitive information. Traditional encryption methods, such as the Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA), have been extensively employed to secure data at rest and in transit. With the growing reliance on cloud services, safeguarding user data has become imperative. Creating a secure environment requires robust protection mechanisms [1]. The AES and RSA are two cornerstone encryption algorithms widely adopted for data protection. AES, a symmetric encryption algorithm, is renowned for its efficiency and speed, making it suitable for encrypting large datasets. Conversely, RSA, an asymmetric encryption algorithm, is primarily utilized for secure key exchanges and digital signatures due to its robust security features. However, these conventional techniques require data to be decrypted before processing, thereby exposing plaintext data during computation and potentially compromising security [2]. This inherent limitation underscores the need for encryption methods that facilitate secure data processing without decryption.

Homomorphic encryption (HE) has emerged as a transformative cryptographic approach, enabling computations to be performed directly on encrypted data without necessitating decryption. This paradigm ensures that data remains confidential throughout the processing lifecycle, thereby mitigating risks associated with data breaches during computation. The applicability of HE spans various domains, including secure data analysis, privacy-preserving machine learning, and confidential financial computations [3]. HE schemes are categorized based on their operational capabilities:

Partially Homomorphic Encryption (PHE): supports a limited set of operations (either addition or multiplication) on ciphertexts.

Somewhat Homomorphic Encryption (SHE): allows both addition and multiplication operations but with constraints on the number of permissible operations before decryption is required.

Fully Homomorphic Encryption (FHE): enables unrestricted addition and multiplication operations on ciphertexts, allowing arbitrary computations on encrypted data.

Within the energy sector, the integration of advanced metering infrastructure and smart grids has led to the generation of vast amounts of consumption data [4]. While this data is instrumental for optimizing energy distribution and consumption, it also raises significant privacy concerns. For instance, detailed energy usage patterns can unconsciously reveal personal habits of consumers. Implementing HE in this context allows for the aggregation and analysis of energy consumption data while preserving individual privacy. A notable application is the anonymous aggregation of production and consumption data in energy communities, facilitating efficient energy management without compromising user confidentiality [5]. The evolution of data security mechanisms has been pivotal in safeguarding sensitive information across various sectors. Within the energy industry, the advent of smart grids and advanced metering infrastructures has necessitated the development of robust encryption techniques to protect consumer data and ensure operational integrity.

The realization of FHE was a significant milestone, as it addressed the limitations of PHE and SHE by permitting complex computations on encrypted data without compromising security. However, FHE schemes are often associated with substantial computational overhead, necessitating ongoing research to enhance their efficiency and practicality.

HE integration in the energy sector addresses privacy concerns in energy data collection and analysis, with key applications including [6].

Privacy-Preserving energy metering: by employing additive homomorphic encryption schemes, utility companies can aggregate energy consumption data without accessing individual readings, thereby preserving consumer privacy [6].

Secure Peer-to-Peer (P2P) energy trading: incorporating HE with blockchain technology facilitates confidential transactions in P2P energy markets. Homomorphically encrypted transactional data stored on the blockchain allows smart contracts to execute functions without revealing sensitive information, enhancing both security and privacy [7].

Enhanced cybersecurity in energy systems: HE enables secure computations on encrypted operational data, mitigating risks associated with cyber threats targeting critical energy infrastructure [8]. This approach ensures that sensitive data remains protected even during processing, thereby enhancing the overall cybersecurity posture of energy systems.

Grid optimization and security: HE can enhance the security of grid operations by enabling secure computations on operational data. This ensures that sensitive information, such as grid load and demand forecasts, remains protected even during processing [9]. Implementing HE in grid management systems can bolster resilience against cyber threats and unauthorized data access.

Despite its security advantages, HE is often criticized for its computational inefficiencies compared to traditional encryption methods. Studies have shown that homomorphic operations incur significant performance overheads, making them slower than symmetric encryption algorithms like AES [10]. For instance, while AES is optimized for fast, secure encryption of large datasets, HE schemes require more computational resources, leading to increased processing times.

A comprehensive assessment conducted by Zhao et al. [11] evaluated four single homomorphic encryption algorithms, analyzing their security characteristics and efficiency. The study revealed that while HE provides enhanced security, it incurs higher computational costs compared to symmetric encryption algorithms like AES and asymmetric algorithms like RSA. Further research by Crihan et al. [12] compared the partially homomorphic encryption algorithm with the Brakerski–Gentry–Vaikuntanathan (BGV) fully homomorphic encryption scheme. The study focused on factors such as encryption time and computational complexity, concluding that while BGV offers comprehensive homomorphic capabilities, it demands greater computational resources, impacting performance .

This paper aims to investigate the efficiency and feasibility of implementing homomorphic encryption in real-world applications, with a particular focus on the electricity and energy sectors. By conducting experimental evaluations on encrypted energy consumption data. This study aims to demonstrate the practical viability of homomorphic encryption for secure data analytics. Additionally, the performance of HE is compared with traditional encryption methods to provide a nuanced understanding of its applicability in the energy industry.

2. Research methods

In this paper, we have studied the reconstruction of a fully homomorphic encryption scheme based on the RLWE (Ring Learning With Errors) problem. RLWE is a structured variant of the LWE problem. RLWE-based HE schemes were introduced to improve efficiency by allowing compact keys and ciphertexts. Moreover, Polynominal ring operations are more efficient than unstructured matrix operations in LWE. Previous studies have applied homomorphic encryption to the energy sector, focusing on privacy-preserving data aggregation and forecasting. For instance, a study demonstrated the use of the Fan-Vercauteren (FV) scheme for secure load forecasting in smart grids [13]. However, RLWE-based HE scheme introduces optimizations specifically for energy data such as efficient handling of time-series consumption data, real-time processing capabilities, or integration with smart grid protocols. Many RLWE-based schemes prioritize security, sometimes at the expense of computational efficiency. Zhang et al. [14] introduced techniques to reduce the complexity of key switching and modulus switching operations, which are critical for improving the performance of FHE schemes. These optimizations make the scheme more practical for real-world applications, including those in the energy sector where processing large volumes of encrypted data efficiently is essential. The homomorphic encryption is a powerful cryptographic approach that allows computations to be

performed directly on encrypted data, ensuring that sensitive information remains protected even during processing. This property addresses a fundamental shortcoming of traditional encryption methods such as AES or RSA, which require data to be decrypted before any meaningful computation can occur, thereby exposing it to potential risks, especially in outsourced or cloud-based environments. In the context of energy systems and smart grids, where massive volumes of consumption data are continuously collected from distributed smart meters and sensors, preserving user privacy while enabling real-time analytics is a critical challenge. HE offers a secure and practical solution by allowing utilities or third-party service providers to perform operations such as data aggregation, load forecasting, and usage profiling directly on encrypted consumption data, without ever accessing the raw values. This makes it possible to deliver intelligent energy services while complying with strict privacy regulations and protecting consumers from data misuse. By extending cryptographic protection into the computation phase, homomorphic encryption not only enhances the security architecture of modern energy infrastructures but also paves the way for trustworthy, privacy-preserving innovation in energy data analytics. As a results, we design an efficient homomorphic encryption scheme requires balancing security, functionality, and performance.

Below is a detailed design of a FHE scheme suitable for electronic data (this schema can also be used for some other data such as smart grids, IoT, finance, etc.), with support for both addition and multiplication. This design builds on lattice-based cryptography using the Ring Learning With Errors (RLWE) problem [15], which is standard in modern HE. *RLWE* is a hard mathematical problem used as the foundation for many lattice-based cryptographic schemes, including homomorphic encryption. It extends the classic Learning With Errors (LWE) problem to polynomial rings, providing better efficiency and compact key sizes. In RLWE, the goal is to distinguish noisy polynomial equations of the form $b = a.s + e\ (mod\ q)$ from random ones, where a, s, e are polynomials and e is a small error term. The presumed hardness of solving RLWE even with quantum computers makes it a strong candidate for post-quantum cryptography.

By utilizing the RLWE problem, we can construct a secure and efficient FHE scheme tailored for encrypting and processing electrical and energy data. Below are the details of the scheme of fully homomorphic encryption.

2.1. Mathematical Preliminaries.

We work within a polynomial ring:

$$R_q = \frac{\mathbb{Z}_q[x]}{f(x)} \tag{1}$$

where: $\mathbb{Z}_q$: integers modulo a large prime or power-of-two modulus q, consisting of the set $\{0,1,2,\dots,q-1\}$ with addition and multiplication performed modulo q. The choice of q influences the security and efficiency of the encryption scheme.

x: an indeterminate or variable used to construct polynomials. Acts as the placeholder for the polynomial terms, enabling the representation of data within the ring structure.

$f(x) = x^n + 1$: a cyclotomic polynomial (when q is a power of 2), which defines the ring structure.

n: the degree of the polynomial $f(x)$, often chosen as a power of two (e.g., 1024, 2048).

R_q: ring of polynomials with coefficients modulo q, degree less than n

This structure enables fast polynomial arithmetic using the Number Theoretic Transform (NTT). NTT serves as a significant mathematical instrument that has gained prominence in the advancement of Post Quantum Cryptography (PQC) and HE. The efficient calculation of polynomial multiplication through the convolution theorem, achieving a quasi-linear complexity of $O(nlogn)$ when utilizing Fast Fourier Transform-style algorithms, has established it as a crucial element in contemporary cryptography [16].

2.2. Key generation

In homomorphic encryption schemes, key generation is a fundamental process that establishes the cryptographic keys necessary for secure encryption and decryption operations. The specifics of this process can vary depending on the particular homomorphic encryption scheme in use. The key generation is defined as follows:

where: n: polynomial degree (power of 2, e.g., 4096 or 8192),

q: ciphertext modulus (large integer, e.g., 2^{50}),

χ: discrete Gaussian or centered binomial error distribution,

s: *secret key*, small random polynomial sampled from χ,

a: random public polynomial sampled uniformly from R_q,

e: noise polynomial sampled from χ,

b: constructed to "hide" the secret s.

Secret key generation:

Select a polynomial s from the ring R_q with coefficients sampled from the error distribution χ. This ensures that s is a "small" polynomial, which is crucial for the security and correctness of the scheme.

Public key generation:

Choose a polynomial a a uniformly at random from R_q. This randomness is essential to ensure the unpredictability of the public key. Then, generate the error polynomial e from the error distribution χ. Like s, this polynomial has small coefficients. Finally, the public key component is calculated by the fomula:

$$b = -a.s + e \ mod \ q \tag{2}$$

This computation ensures that the public key is related to the secret key in a way that maintains the hardness assumptions of the RLWE problem. Therefore, the security key pair for the FHE scheme will consist of

Secret key (sk): the polynomial s, which must be kept confidential as it is used for decryption.

Public key (pk): the pair (a, b), which can be openly shared and is used for encryption.

2.3. Encryption and Decryption process

Encryption:

The objective of encryption in FHE scheme transform plaintext data into ciphertext, allowing computations to be performed without exposing the original information. The encryption steps are as follows:

The first step, the data to be encrypted, plaintext (m), represented as a polynomial in the plaintext ring, where, $m \in R_t$, $t \ll q$, (t is a smaller integer representing the coefficient space for plaintexts, commonly $t = 2$ for binary messages or $t = 256$ for byte-sized messages). The space of plaintext will be calculated by the formula:

$$R_t = \frac{\mathbb{Z}_t[x]}{x^n + 1} \tag{3}$$

In the second step, generate random polynomials (u , e_1, e_2), sampled from a discrete Gaussian distribution χ to introduce necessary randomness and ensure security.

Then, scale m to match the ciphertext modulus q , resulting in:

$$m' = \left\lfloor \frac{q}{t}.m \right\rfloor \in R_q \tag{4}$$

The final step is the computation of the ciphertext components to finish the encryption process, using the public key components (a, b), compute the ciphertext as a pair of polynomials, by the fomulas as follows:

$$c_0 = b \cdot u + e_1 + m' \ mod \ q \tag{5}$$

$$c_1 = a \cdot u + \ e_2 \ mod \ q \tag{6}$$

Ciphertext $(c_0, c_1) \in R_q^2$, which can be publicly shared and used for homomorphic computations. The encrypted message is now hidden within the structure, and no one without the secret key can extract it due to the RLWE hardness.

Decryption:

Decryption Process and provide a concrete example to illustrate both encryption and decryption steps. Its objective retrieve the original plaintext from the ciphertext using the secret key. The decryption steps are as follows:

Compute the inner product by calculating the polynominal $m'(x)$ as:

$$m'(x) = c_0(x) + c_1(x).s(x) \, mod \, q \tag{7}$$

This operation combines the ciphertext components with the secret key to produce an intermediate polynomial that contains the scaled plaintext and associated noise.

Recover the plaintex by scaling down $m'(x)$ to the plaintext modulus t and round to the nearest integer, $m(x)$ calculate by the fomula:

$$m(x) = \left\lceil \frac{t}{q}.m'(x) \right\rfloor \, mod \, t \tag{8}$$

This step maps the coefficients of $m'(x)$ back to the original plaintext space, effectively removing the scaling applied during encryption.

3. Results and Discussions

In the electric power industry, the integrity and confidentiality of electrical data are paramount. This data encompasses real-time grid operations, consumption metrics, and infrastructure diagnostics, serving as the backbone for efficient energy distribution and management. Unauthorized access or manipulation of this information can lead to severe consequences, including service disruptions, financial losses, and compromised public safety. As the sector increasingly integrates digital technologies, the attack surface for cyber threats expands, necessitating robust data protection mechanisms [17].

Fully Homomorphic Encryption (FHE) offers a compelling solution by enabling computations on encrypted data without requiring decryption. This ensures that sensitive electrical data remains confidential even during processing, effectively mitigating risks associated with data exposure to untrusted environments or external service providers. By implementing FHE, electric power companies can perform essential analyses such as load forecasting, fault detection, and demand response optimization on encrypted datasets, preserving data privacy and integrity. This approach not only enhances cybersecurity resilience but also aligns with regulatory compliance and bolsters stakeholder trust in an era where data breaches are increasingly prevalent.

3.1. Processing model

Electrical meters are tools used to measure how much power is used by consumers. In addition to being required for utility billing and electrical grid management, the information gathered by these meters may be utilized to evaluate the environmental effect of buildings. Previous studies have shown that by identifying variations in energy use, unsecured metering data may be utilized to deduce certain information about building residents' actions. A period of low electricity use, for instance, might indicate that the building's occupants are not there. Data privacy safeguards for metering data that do not impair the availability and quality of data used for energy management and billing applications are becoming more and more necessary as smart metering becomes more widespread.

This section presents a four-stage model illustrating how The FHE can be applied to the secure processing of electrical energy consumption data in a smart building as shown in Figure 1. The aim is to calculate the average energy consumption across multiple measurement points (referred to as sites) without ever revealing the raw data values during the computation process. These raw measurements must be protected to ensure privacy and prevent unauthorized access.

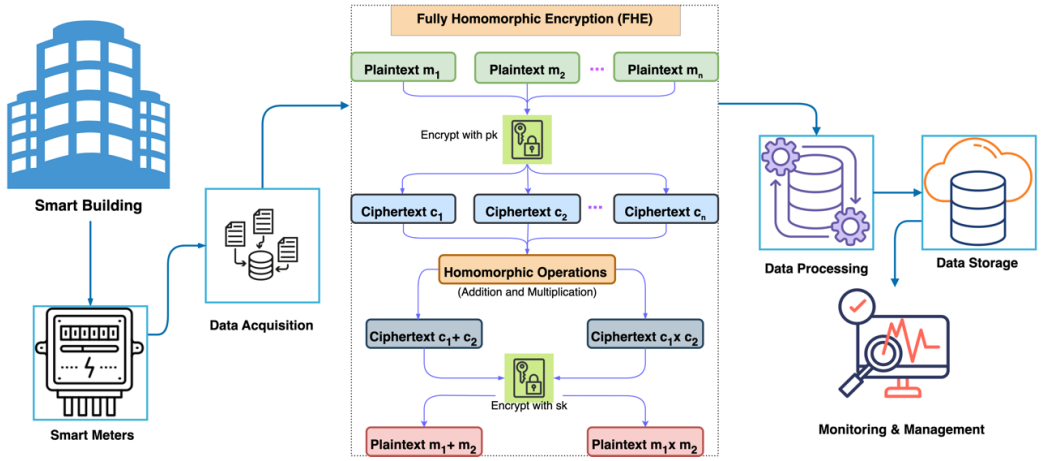

Figure 1. *Data processing model and full homomorphic encryption for electric data*

3.2. Data acquisition

From the diagram in Figure 1, the process of collecting, processing and storing electricity consumption data in smart buildings is carried out as follows: electricity measuring devices such as smart meters are deployed at the main inlet meter located in areas (such as floors, rooms, functional areas), large equipment (such as central air conditioning, elevators, pumps...). The included sensors may include current sensor, voltage sensor, power sensor, and ambient temperature sensor. The collected data includes Electricity consumption (kWh), Current (A), Voltage (V), Real-time power consumption (kW), and Power Factor. The acquisition process is handled through a component called *Data Acquistion*, where it is aggregated and formatted into structured digital representations, referred to as plaintext values (e.g., m_1, m_2, m_3). The measuring devices use data transmission protocols, including wired (Modbus, BACnet, KNX), and wireless (ZigBee, LoRaWAN, Wi-Fi, Bluetooth LE) options. The collection process is handled through a component called Data Acquistion, measuring devices use data transmission protocols: wired: Modbus, BACnet, KNX, wireless: ZigBee, LoRaWAN, Wi-Fi, Bluetooth LE. The data collector will collect data from measuring devices, then synchronize and standardize the data format. Depending on the sensitivity of the data, it will be collected periodically (1 minute/ 1 time, or 5 minutes/ 1 time, or on request).

3.3. Data encryption and decryption

To ensure privacy and data security from the outset, each plaintext measurement is encrypted using a public key (pk) within the FHE component. This produces encrypted data (ciphertexts c_1, c_2, c_3), which can be safely processed without ever exposing the original plaintext values. Significantly, homomorphic operations such as addition and multiplication are then performed directly on the encrypted data, enabling meaningful computation like total energy usage or power comparisons without requiring decryption.

The resulting encrypted outputs (e.g., $c_1 + c_2$ or $c_1 \times c_2$) are later decrypted using the corresponding private key (sk) to reveal accurate, privacy-preserved computation results (e.g., $m_1 + m_2$, $m_1 \times m_2$). After decryption, the data flows to the Data Processing component, where advanced analytics are performed, such as trend analysis, predictive modeling, or peak load detection.

Processed results are then transferred to a Data Storage system, typically a secure cloud platform or on-premise server, which archives the data for future audits and longitudinal analysis. Finally, all processed insights are displayed through a Monitoring and Management interface, where facility managers can track real-time consumption, receive alerts for anomalies, and make informed decisions regarding energy optimization.

By integrating FHE into the data pipeline, this processing provides end-to-end encryption, ensuring data privacy even during computation making it an ideal solution for secure and intelligent energy management in modern buildings.

3.4. Experimental testbed

To illustrate the experimental process for the fully homomorphic encryption scheme, the electricity dataset shown below was utilized. In order to evaluate the homomorphic encryption method, the following table of fundamental data from the electricity and energy industry contains information on energy consumption, electricity production, and basic characteristics, as shown in Table 1. We also use a library called pyFHE (Python for Homomorphic Encryption Libraries) [18] to conduct experiments on our electricity consumption dataset using FHE, pyFHE is a suitable Python library that provides a user-friendly interface for homomorphic encryption operations. The pyFHE is designed to bring FHE capabilities to Python, allowing computations on encrypted data without decryption. It wraps functionalities from multiple FHE libraries, such as Microsoft's SEAL[19], providing a syntax similar to standard arithmetic operations in Python. This makes it accessible for both demonstrations and complex applications like machine learning algorithms.

Table 1. *Basic data sheet on electricity and energy sector*

ID	Date	Power Plant	Energy Type	Output (MWh)	Consumption (MWh)	Selling Price (VND/MWh)	CO_2 Emissions (tons)
1	01/01/2025	Power Plant A	Thermal	1200	1100	1,200,000	300
2	02/01/2025	Power Plant B	Hydropower	850	800	900,000	0
3	03/01/2025	Power Plant C	Solar	500	450	1,100,000	0
4	04/01/2025	Power Plant D	Wind	600	550	1,000,000	0
5	05/01/2025	Power Plant E	Thermal	1300	1250	1,300,000	350
6	06/01/2025	Power Plant F	Hydropower	900	850	950,000	0
7	07/01/2025	Power Plant G	Solar	600	550	1,150,000	0
8	08/01/2025	Power Plant H	Thermal	1400	1350	1,400,000	400
9	09/01/2025	Power Plant I	Wind	700	650	1,050,000	0
10	10/01/2025	Power Plant J	Hydropower	800	750	920,000	0

Data field descriptions:

ID: Unique identifier for each data record.

Date: Date when the data was recorded.

Power plant: Name of the power plant.

Energy type: Type of energy used (e.g., thermal, hydropower, solar, wind).

Output (MWh): Total electricity generated in megawatt-hours.

Consumption (MWh): Total electricity consumed in megawatt-hours.

Selling price (VND/MWh): Average selling price of electricity per megawatt-hour, in Vietnamese Dong.

CO_2 emissions (tons): Amount of CO_2 emissions produced during electricity generation (applicable to thermal plants only).

With the above scheme and the data table in Table 1, To advance the encryption process, appropriate parameters are carefully selected at this stage. The encryption and decryption processes, based on the RLWE (Ring Learning With Errors) model, are explained in detail in accordance with the FHE scheme and specifically applied to two columns: Output and Selling Price in the data table.

Input data for the FHE scheme as the follows:

Encryption: Output = 1200 (MWh), Selling price = 1,200,000 (VND/MWh). Then calculate: Revenue = Output × Selling price on encrypted data, Finally: decrypt the result to clear form.

Table 2 illustrates some selected parameters for the encryption process:

In the encrytion and decrytion stages, we assume the data that needs to be encrypted is messages $m_1 = 1200$ (O$utput\ of\ production$ MWh), $m_2 =$ 1,200,000 (Selling price per MWh).

Table 2. *Select encryption parameters*

Parameter	Example	Description
n	4096	Polynomial degree, defines ring: $R_q = \mathbb{Z}_q[x]/(f(x))$ We chose n=4096 to provide a good balance between speed and accuracy, used in the demo.
q	$2^{50} \approx 1.1259 \times 10^{15}$	Ciphertext modulus, a large integer (In FHE, the larger q is, the wider the noise budget is to allow for deeper computations.)
χ	Centered binomial (±1) or discrete Gaussian	Noise distribution (randomly generate coefficients {−1, 0, +1} with uniform probability. The goal of this parameter is to keep the value small enough to ensure that the ciphertext does not exceed the modulus q limit, but large enough to ensure security against solving linear systems of equations.)
s	Small random polynomial from χ	*Secret key* (s=[1,0,−1,1,0,...,0]∈ R_q)
a	Random public polynomial from R_q	Public coefficient (Random in [0, q))
e	Small noise polynomial from χ	Adds obfuscation
b	Computed to hide s: $b = -a.s + e\ mod\ q$	Calculated to generate public key

Encryption:

Encypt $m_1, m_2 \in \mathbb{Z}_q$ into polynomial form.

Sample random polynomials: $(u, e_1, e_2) \leftarrow \chi$

Compute ciphertext for m_1, m_2:

$$Enc\ (m_1) = (c_{01}, c_{11}) \quad (9) \qquad Enc\ (m_2) = (c_{02}, c_{12}) \quad (10)$$

Where (c_{01}, c_{11}) and (c_{02}, c_{12}) is calculated:

$$c_{01} = b \cdot u + e_1 + m_1\ mod\ q \quad (11) \qquad c_{11} = a \cdot u + e_2\ mod\ q \quad (13)$$

$$c_{02} = b \cdot u + e_2 + m_2\ mod\ q \quad (12) \qquad c_{12} = a \cdot u + e_2\ mod\ q \quad (14)$$

To compute the encrypted *Revenue = Output of production × Selling price*, multiplication is carried out on the encrypted copy, as demonstrated below:

$$Enc\ (m_1, m_2) = (c'_0, c'_1, c'_2) \quad (15)$$

where:

$$c'_0 = c_{01}.c_{02} \quad (16) \qquad c'_1 = c_{01}.c_{12} + c_{11}.c_{02} \quad (17) \qquad c'_2 = c_{11}.c_{12} \quad (18)$$

Decryption: From encrypted result (c_{01}, c_{11}) and (c_{02}, c_{12}). The decryption stage is calculated according to the following formula:

$$m' = c'_0 + c'_1.s\ mod\ q \quad (19)$$

If noise is small and parameters are well-chosen:

$$m' = m_1.m_2 = 1200 \times 1{,}200{,}000 = 1{,}440{,}000{,}000$$

Table 3 summarizes the resulting process of encryption and decryption.

Table 3. *Table summarizes the resulting process of encryption and decryption*

Field	Plaintext	Encrypted (FHE)	Operation
Production	1200 (MWh)	$Enc\ (1200) = (c_{01}, c_{11})$	Encrypted input
Price	1,200,000 (VND/MWh)	$Enc\ (1{,}200{,}000) = (c_{02}, c_{12})$	Encrypted input
Revenue	-	Multiply ciphertexts	Homomorphic multiplication
Result	-	$Decrypt(c'_0 + c'_1.s) = 1{,}440{,}000{,}000$	Decryption

Encryption and decryption time:

In this comparative study, the performance characteristics of three prominent encryption schemes FHE, RSA, and AES are evaluated based on time-based metrics across multiple computational rounds. A time series analysis is employed to capture variations in encryption time and decryption time over successive encryption rounds and decryption rounds, where each round

represents a consistent execution of the cryptographic operation under varying noise, key, or system states. FHE exhibits the highest computational overhead due to polynomial-based operations and noise management, yielding encryption times exceeding 2000 milliseconds and ciphertexts in the range of several megabytes. RSA, while asymmetrical and moderately efficient, performs within 200–300 milliseconds per operation. AES, being a lightweight symmetric cipher, achieves the lowest latencies (~50 ms) and minimal ciphertext sizes (~16 KB), the results showing in Figure 2.

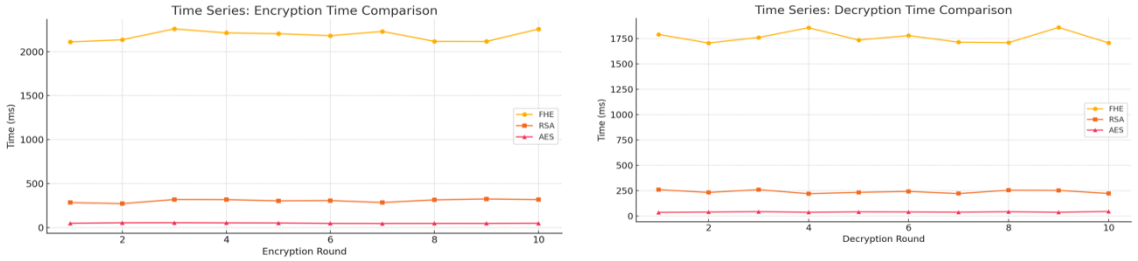

Figure 2. *Time of encryption and decryption*

Security lever and Computational capability:

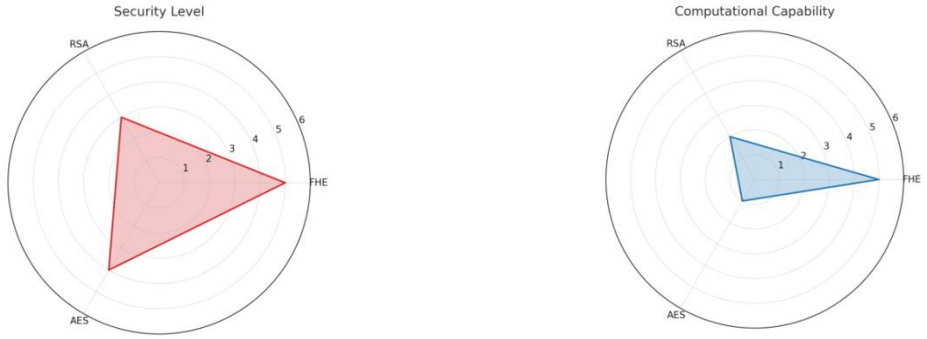

Figure 3. *Comparison of security lever and computational capability*

In Figure 3, in order to compare encryption algorithms effectively, we adopt a 1–6 rating scale for two key criteria: Security Level and Computational Capability. This scale offers sufficient granularity to distinguish between traditional schemes like RSA and AES and FHE. For Security Level, a score of 1 represents minimal protection, vulnerable to modern attacks. Scores 4–5 reflect strong classical cryptographic security (e.g., AES), while a score of 6 denotes high resilience against both classical and quantum threats, typically achieved by lattice-based or FHE schemes. For Computational Capability, the scale measures how well an algorithm supports operations on encrypted data. A score of 1 indicates no support (e.g., AES), while 2–3 reflect limited homomorphic properties (e.g., RSA). Scores 4–5 denote partial homomorphic encryption that supports either addition or multiplication. A full score of 6 is assigned to FHE, which allows arbitrary computations on encrypted data without decryption.

4. Conclusions

Homomorphic encryption represents a significant advancement in cryptographic research, offering the unique capability to perform computations on encrypted data without decryption. Its application in the energy and electricity domains holds promise for enhancing data privacy and security. In this paper, This work presents a fully homomorphic encryption scheme designed to protect sensitive information within the electricity and energy domain. The homomorphic encryption method is also compared with other conventional encryption techniques, such as RSA and AES, to highlight its advantages and limitations. Additionally, this method is applied and tested on an electric energy consumption dataset from smart buildings equipped with numerous controllers and sensors. The results also demonstrate that the privacy and security as well as the capability to calculate on encrypted data is quite good, and the ability for data to be absolutely

secured even when direct managers obtain this data. especially when the data is stored on a third party or in a cloud computing environment. However, FHE currently has limitations in performance and ciphertext size, encryption and decryption speed. Ongoing research and development efforts are directed toward optimizing homomorphic encryption schemes to address this problem more effectively. Moreover, in the context of increasingly growing big data, FHE promises to become a foundational technology to help ensure privacy and security in distributed computing, cloud computing, and artificial intelligence when applying real-time applications.

REFERENCES

[1] S. Fatima *et al.*, "Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing," *Eng. Proc.*, vol. 20, no. 1, Art. no. 1, Jul. 2022, doi: 10.3390/engproc2022020014.

[2] U. Mattsson, "Security and Performance of Homomorphic Encryption," Global Security Mag Online, 2021. [Online]. Available: https://www.globalsecuritymag.com/Security-and-Performance-of,20210601, 112333.html. [Accessed Mar. 23, 2025].

[3] S. Z. E. Mestari, G. Lenzini, and H. Demirci, "Preserving data privacy in machine learning systems," *Comput. Secur.*, vol. 137, Feb. 2024, Art. no. 103605, doi: 10.1016/j.cose.2023.103605.

[4] S. Burtner, "Homomorphic Encryption for Electrical Metering Aggregation: Protecting the Privacy of Building Tenants," Pacific Northwest National Laboratory, Sep. 2024.

[5] D. Strepparava, F. Rosato, L. Nespoli, and V. Medici, "Privacy and Auditability in the Local Energy Market of an Energy Community with Homomorphic Encryption," *Energies*, vol. 15, no. 15, Art. no. 15, Jul. 2022, doi: 10.3390/en15155386.

[6] F. D. Garcia and B. Jacobs, "Privacy-Friendly Energy-Metering via Homomorphic Encryption," in *Security and Trust Management*, J. Cuellar, J. Lopez, G. Barthe, and A. Pretschner, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 226–238.

[7] D. Mitrea *et al.*, "Smart contracts and homomorphic encryption for private P2P energy trading and demand response on blockchain," *Heliyon*, vol. 9, no. 11, Nov. 2023, doi: 10.1016/j.heliyon.2023.e22357.

[8] S. Abdelkader *et al.*, "Securing modern power systems: Implementing comprehensive strategies to enhance resilience and reliability against cyber-attacks," *Results Eng.*, vol. 23, Sep. 2024, Art. no. 102647, doi: 10.1016/j.rineng.2024.102647.

[9] B. Li, M. Xu, Y. Zhou, H. Liu, and R. Zhang, "Optimization of Security Identification in Power Grid Data through Advanced Encryption Standard Algorithm," *J. Cyber Secur. Mobil.*, vol. 13, no. 02, pp. 239–264, Feb. 2024, doi: 10.13052/jcsm2245-1439.1323.

[10] N. Kshetri *et al.*, "algoTRIC: Symmetric and asymmetric encryption algorithms for Cryptography - A comparative analysis in AI era," *arXiv*: *arXiv:2412.15237*, Dec. 12, 2024, doi: 10.48550/arXiv.2412.15237.

[11] M. Zhao E and Y. Geng, "Homomorphic Encryption Technology for Cloud Computing," *Procedia Comput. Sci.*, vol. 154, pp. 73–83, Jan. 2019, doi: 10.1016/j.procs.2019.06.012.

[12] G. Crihan *et al.*, "A Comparative Assessment of Homomorphic Encryption Algorithms Applied to Biometric Information," *Inventions*, vol. 8, no. 4, Aug. 2023, Art. no. 4, doi: 10.3390/inventions8040102.

[13] J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren, "Privacy-friendly Forecasting for the Smart Grid using Homomorphic Encryption and the Group Method of Data Handling," 2016. [Online]. Available: https://eprint.iacr.org/2016/1117. [Accessed May 04, 2025].

[14] X. Zhang, C. Xu, C. Jin, R. Xie, and J. Zhao, "Efficient fully homomorphic encryption from RLWE with an extension to a threshold encryption scheme," *Future Gener. Comput. Syst.*, vol. 36, pp. 180–186, Jul. 2014, doi: 10.1016/j.future.2013.10.024.

[15] O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," *arXiv*: *arXiv:2401.03703*, Jan. 08, 2024, doi: 10.48550/arXiv.2401.03703.

[16] A. Satriawan *et al.*, "Conceptual Review on Number Theoretic Transform and Comprehensive Review on Its Implementations," *IEEE Access*, vol. 11, pp. 70288–70316, 2023, doi: 10.1109/ACCESS.2023.3294446.

[17] T. Krause, R. Ernst, B. Klaer, I. Hacker, and M. Henze, "Cybersecurity in Power Grids: Challenges and Opportunities," *Sensors*, vol. 21, no. 18, Sep. 2021, Art. no. 6225, doi: 10.3390/s21186225.

[18] S. Erabelli, "pyFHE - a Python library for fully homomorphic encryption," M. E. Thesis, Massachusetts Institute of Technology, 2020. [Online]. Available: https://dspace.mit.edu/handle/1721.1/129204. [Accessed Apr. 01, 2025].

[19] W. Redmond, "Microsoft SEAL: Fast and Easy-to-Use Homomorphic Encryption Library," Microsoft Research, 2023. [Online]. Available: https://www.microsoft.com/en-us/research/project/microsoft-seal/. [Accessed May 14, 2025].