International Journal of Safety and Security Engineering
Vol. 15, No. 10, October, 2025, pp. 2155-2167
Journal homepage: http://iieta.org/journals/ijsse

Secure Data Encryption in Energy Production and Management Systems: Integrating Chaos Bifurcation and Polynomial High Order Fibonacci for Enhanced Cybersecurity

Tulus[1*], Jonathan Liviera Marpaung[1], Syafrizal Sy[2], Kiki Ariyanti Sugeng[3], Rinovia Simanjuntak[4], Suriati[5]

[1] Department of Mathematics, Universitas Sumatera Utara, Medan 20155, Indonesia
[2] Department of Mathematics, Universitas Andalas, Padang 25163, Indonesia
[3] Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia
[4] Department of Mathematics, Institut Teknologi Bandung, Bandung 40132, Indonesia
[5] Department of Informatics, Universitas Harapan Medan, Medan 20153, Indonesia

Corresponding Author Email: tulus@usu.ac.id

https://doi.org/10.18280/ijsse.151018

Received: 11 September 2025
Revised: 12 October 2025
Accepted: 23 October 2025
Available online: 31 October 2025

Keywords:
PHOF, chaotic maps, data encryption, cybersecurity, energy management

ABSTRACT

Secure data handling is paramount in energy production and management systems, where cyber threats pose significant risks to operational continuity. In response, this study proposes an integration of chaos bifurcation and the Polynomial High Order Fibonacci (PHOF) approach to fortify encryption protocols in critical energy infrastructures. The method combines polynomial-based Fibonacci sequences with chaotic iteration steps analyzed through bifurcation to generate non-linear keystreams. These keystreams deliver robust confusion and diffusion capabilities, effectively mitigating brute-force and statistical attacks. Experimental findings confirm substantial gains in randomness, validated by entropy assessments and avalanche effect tests. Moreover, chaos bifurcation analysis highlights the sensitivity of the system's chaotic parameters, reinforcing security under varying conditions. Despite these layered mechanisms, the PHOF-chaotic scheme maintains a low computational burden, making it highly suitable for real-time data exchange within energy monitoring and control frameworks. Consequently, coupling PHOF with chaos bifurcation techniques significantly strengthens cybersecurity for energy systems, ensuring both reliable performance under operational demands and resilient protection against evolving cyber threats.

1. INTRODUCTION

The digital era has brought significant advancements in technology, transforming how data is created, stored, and transmitted. However, this rapid evolution has also introduced complex challenges in maintaining data security. One of the primary concerns is the increasing volume of data generated every second, ranging from personal information and financial transactions to corporate secrets and governmental records. This vast amount of data attracts cybercriminals seeking to exploit vulnerabilities for financial gain, espionage, or disruption [1, 2]. Cyber threats have become more sophisticated over time, including phishing, ransomware, malware, and denial-of-service attacks. Phishing attacks trick users into providing sensitive information, while ransomware encrypts data, demanding payment for its release. Malware can infiltrate systems to steal, corrupt, or destroy data, and denial-of-service attacks overwhelm networks, rendering them inoperable. The dynamic nature of these threats poses continuous challenges for cybersecurity experts. The proliferation of Internet of Things (IoT) devices has further complicated data security. Many IoT devices lack robust security features, making them easy targets for hackers. These devices, when compromised, can serve as entry points to larger networks, endangering sensitive data. Additionally, cloud computing, while offering scalable storage solutions, introduces risks related to data breaches, unauthorized access, and data loss. One of the critical challenges is the constant need for encryption methods that can withstand evolving threats [3]. While traditional encryption techniques such as RSA and AES remain widely used, they are increasingly being scrutinized due to potential vulnerabilities posed by future quantum computing advancements. This has led to the exploration of new encryption models, such as Fibonacci sequences and polynomial functions, which offer promising alternatives in the face of these emerging threats. This has spurred research into new encryption models, such as those based on Fibonacci sequences and polynomial functions, which offer promising alternatives due to their complexity and unpredictability. Human factors also contribute to data security challenges. Weak passwords, lack of awareness, and negligence in following security protocols make systems vulnerable. Organizations often face difficulties in ensuring that all employees adhere to best practices for data security, leading to potential breaches. Moreover, regulatory challenges add another layer of complexity. Data protection laws vary

across regions, making it difficult for multinational organizations to comply with different standards simultaneously. Non-compliance can result in hefty fines and reputational damage. To address these challenges, continuous innovation in encryption methods is essential. Integrating Polynomial High Order Fibonacci (PHOF) models into encryption algorithms can enhance data security by providing more complex and resilient encryption keys. This method leverages the unpredictable nature of high-order polynomial sequences, making it harder for attackers to crack the encryption. The digital era's rapid technological advancements have made data security a critical concern. Sophisticated cyber threats, IoT vulnerabilities, human errors, and regulatory challenges necessitate the development of advanced encryption techniques. PHOF-based encryption presents a promising solution, significantly enhancing data security by introducing higher complexity into the key generation process. By replacing the linear recurrence of standard Fibonacci sequences with higher-order polynomial terms, the system generates more unpredictable and non-linear sequences. This increased complexity makes the key space exponentially larger and more resistant to cryptanalysis, ensuring that data remains secure even against evolving cyber threats.

Conventional encryption methods such as RSA, AES, and DES have long been the backbone of data security. However, as technology evolves, these methods face increasing challenges. One significant limitation is their vulnerability to quantum computing, which can potentially break classical encryption algorithms through quantum algorithms like Shor's algorithm. Additionally, conventional methods often rely on key management systems that can become weak points if not handled securely. Another limitation is the computational overhead of traditional encryption methods, especially when applied to large datasets or real-time communications. High processing power requirements can lead to inefficiencies in resource-constrained environments, such as IoT devices or embedded systems. Moreover, conventional encryption methods can be susceptible to side-channel attacks, where attackers exploit information leaked during the encryption process, such as timing information, power consumption, or electromagnetic leaks. Brute force attacks, although time-consuming, remain a threat as computational power increases. There is also the challenge of maintaining key integrity and distribution. Public key infrastructures (PKI) are complex and require constant monitoring and updating, posing operational challenges. To address these limitations, new approaches are necessary. PHOF-based encryption offers an innovative solution by introducing more complex key generation mechanisms. The use of polynomial sequences adds unpredictability and robustness, making it harder for attackers to decipher encrypted data. Additionally, integrating noise with Fibonacci sequences provides an extra layer of security. This approach not only enhances security but also offers computational efficiency. By leveraging mathematical properties of polynomials and Fibonacci numbers, the encryption process can be optimized for various applications, including IoT, cloud computing, and secure communications. The need for new encryption approaches is evident in the face of emerging threats and technological advancements. PHOF-based encryption presents a promising avenue to enhance data security [4-7].

Fibonacci sequences have been utilized in data encryption due to their unique mathematical properties and inherent complexity. The Fibonacci sequence, where each number is the sum of the two preceding ones, offers an unpredictable pattern that can be exploited for encryption purposes. In previous encryption methods, Fibonacci sequences were used to generate encryption keys, transform data structures, and introduce controlled randomness in the encryption process. While some chaotic encryption methods and Fibonacci-based encryption approaches face limitations, including constrained key spaces and predictability, other methods have been developed to overcome these challenges. However, there is still room for improvement in expanding key spaces and enhancing unpredictability, which our PHOF method aims to address. Many existing methods rely on first-order Fibonacci recursions, which, despite offering some level of unpredictability, can be easily compromised by modern cryptanalysis tools due to their linearity and limited key space. Moreover, typical chaotic systems may not generate sufficiently random keystreams for high-security applications. Our proposed PHOF encryption method addresses these challenges by introducing higher-order polynomial modifications to the Fibonacci recurrence, significantly expanding the key space and increasing the randomness of the generated sequences. This combination of polynomial complexity and chaotic behavior not only enhances the unpredictability of the keystream but also offers greater resistance to both classical and emerging cryptographic attacks, such as those from quantum computing. The novelty of PHOF lies in this integration of chaos and polynomial sequences, which provides a more robust and scalable encryption solution for modern cryptographic needs. The use of Fibonacci graphs further enhanced encryption by leveraging graph theory to create intricate data representations that are difficult to decipher without the correct key. Fibonacci-based encryption has shown promise in adding layers of security through noise integration and complex key generation. However, previous methods faced limitations such as scalability issues and vulnerability to advanced cryptanalysis techniques. PHOF sequences aim to address these challenges by introducing higher-order polynomial functions to generate Fibonacci-like sequences, thereby increasing complexity and security. This section explores various studies that have applied Fibonacci sequences in cryptographic algorithms, highlighting their strengths and limitations and setting the stage for the proposed research. Energy systems, particularly those involved in critical infrastructure like SCADA systems, face unique security challenges due to the need for real-time data processing, the sensitivity of operational data, and the growing risk of remote cyberattacks. Energy systems, like many other critical infrastructures, face unique security challenges due to their real-time data processing needs, the sensitivity of operational data, and the growing risk of remote cyberattacks. These challenges highlight the importance of robust and efficient encryption solutions tailored to meet specific operational demands. Our proposed PPHOF-based encryption addresses these challenges by providing high-security encryption with low computational overhead, making it well-suited to protect sensitive data and ensure operational continuity in energy systems. This approach also aligns with industry regulations that mandate strong cybersecurity measures for critical infrastructure.

PHOF sequences enhance encryption security through the generation of highly complex and non-linear numerical sequences, which are inherently resistant to prediction and cryptographic attacks. By utilizing high-order polynomial functions, the resulting Fibonacci-like sequences possess an

exponential growth in key space, making brute force decryption attempts computationally infeasible. Additionally, the integration of controlled noise within these sequences further obscures the encrypted data, thereby mitigating the risk of pattern recognition by attackers. This method not only amplifies the robustness of encryption keys but also ensures greater data integrity and confidentiality. The application of PHOF in encryption leverages advanced mathematical principles to provide a secure, scalable, and efficient cryptographic solution. Implemented through Python, this approach offers flexibility and performance optimization, rendering it suitable for modern data security challenges. This section presents a detailed exploration of the theoretical underpinnings, cryptographic strength, and implementation advantages of PHOF in enhancing encryption security, adhering to an academic and research-oriented standard.

Over the past decades, secure data encryption has evolved significantly, driven by the increasing need to protect multimedia and sensitive information in diverse applications. Traditional encryption algorithms such as DES, AES, and RSA have long served as the backbone for data security; however, their performance in encrypting high-redundancy data like images has often been limited due to inherent pixel correlations and large data volumes. Researchers have thus turned to chaos theory and novel mathematical constructs to design more robust and efficient encryption schemes. Chaotic systems have attracted substantial attention in cryptographic research because of their inherent properties, namely, high sensitivity to initial conditions, ergodicity, and pseudorandom behavior. These attributes render chaotic maps, such as the logistic map, piecewise linear chaotic maps (PWLCM), and higher-dimensional chaotic systems, particularly suitable for generating complex keystreams [8, 9]. Many studies have demonstrated that integrating chaotic maps into encryption schemes significantly enhances confusion and diffusion, essential features that thwart differential and statistical attacks [10, 11]. For instance, various image encryption methods have utilized chaotic-based frameworks that scramble pixel positions and diffuse pixel values through nonlinear transformations, as evidenced in works employing the Arnold cat map, Baker map, and even hyperchaotic systems. Parallel to chaos-based approaches, Fibonacci sequences have also been investigated for cryptographic applications. While previous chaotic encryption methods, such as those cited in the previous studies [8-11], have successfully utilized chaotic maps for data encryption, our approach introduces a novel modification by applying high-order polynomials to the Fibonacci recurrence. This enhancement significantly increases the key space and randomness of the generated keystream, making it more resistant to brute-force and cryptanalytic attacks. Furthermore, our approach enhances traditional chaotic encryption schemes by integrating high-order polynomial modifications into the Fibonacci sequence, which significantly expands the key space and randomness. This novel combination aims to address existing challenges and provide stronger protection against cryptographic attacks. This dual-layer mechanism ensures stronger protection against modern cryptographic threats, including side-channel attacks and quantum computing challenges. Traditional Fibonacci-based encryption schemes rely on the recursive nature of the Fibonacci sequence to generate keystreams with a degree of unpredictability. However, these methods often suffer from a limited key space and predictable patterns when confined to first-order recurrences. To overcome these limitations, researchers have recently proposed the use of high-order polynomial Fibonacci sequences, which extend the classical Fibonacci recursion by incorporating polynomial modifications [12, 13]. This innovation exponentially expands the key space and introduces an additional layer of complexity, making brute-force and cryptanalytic attacks increasingly infeasible. Chaos and polynomial methods, noise modification has emerged as a promising strategy to further obscure plaintext information [14, 15]. By introducing controlled noise into the encryption process, one can disrupt any residual statistical structure that might otherwise be exploited by attackers. The combination of noise with chaotic systems has shown improved diffusion characteristics, ensuring that even a minor change in the input or encryption parameters yields a vastly different ciphertext. Despite these advancements, a notable research gap remains in the integration of PHOF sequences with chaos-based techniques, especially in the context of energy production and management systems [16, 17]. While many chaotic encryption schemes have been applied to multimedia data, few have explored the synergistic potential of combining high-order polynomial recurrences with chaotic maps and noise modification. The literature indicates that although chaos-based algorithms offer excellent key randomness and sensitivity, the majority of existing approaches do not fully leverage the additional complexity afforded by polynomial modifications. Moreover, the dynamic adaptation of encryption parameters through noise modification remains underexplored, limiting the robustness of many proposed schemes [18-20].

PHOF sequences improve encryption security by generating complex, non-linear sequences that are difficult to predict or break. Unlike linear sequences, high-order polynomials create multifaceted encryption keys with increased randomness, making brute force attacks nearly impossible [7, 21, 22]. This method enhances security by expanding the key space exponentially, integrating controlled noise, and reducing patterns in encrypted data. Additionally, its implementation in Python allows for efficient computation and scalability. In about 200 words, this section emphasizes the advantages of using PHOF in modern encryption, highlighting its potential to offer robust security, computational efficiency, and resistance to evolving cyber threats [23-25]. PHOF sequences improve encryption security by generating complex, non-linear sequences that are difficult to predict or break. Unlike linear sequences, high-order polynomials create multifaceted encryption keys with increased randomness, making brute force attacks nearly impossible. This method enhances security by expanding the key space exponentially, integrating controlled noise, and reducing patterns in encrypted data. Additionally, its implementation in Python allows for efficient computation and scalability [26-29]. This section explores the mathematical foundation, security benefits, and computational advantages of this approach.

Our research addresses this gap by proposing an innovative encryption algorithm that integrates PHOF sequences with chaotic bifurcation analysis and noise modification. This hybrid approach not only enhances the key space but also ensures that the keystream exhibits high entropy and strong sensitivity to initial conditions, qualities that are critical for countering modern cyber threats. By analyzing bifurcation diagrams, our method fine-tunes chaotic parameters to achieve optimal unpredictability, while controlled noise injection further disrupts potential patterns in the ciphertext. The objective includes creating an algorithm that offers high

entropy, non-linearity, and resistance to modern cryptographic attacks such as brute force, side-channel attacks, and quantum computing threats. Furthermore, this research seeks to implement the algorithm using Python, ensuring computational efficiency, scalability, and flexibility across various applications. The study will conduct rigorous simulations to evaluate the algorithm's performance, security, and practical viability in real-world scenarios, contributing significantly to the field of data security through novel encryption techniques.

1.1 Novelty of the paper

The novelty of this research lies in the combined application of PHOF sequences and chaotic iteration for data encryption in energy production and management systems, a domain in which such techniques are not widely explored. Unlike standard Fibonacci-based methods, the high-order polynomial modifications significantly broaden the key space, making brute-force attacks infeasible. Furthermore, the integration of controlled noise and chaotic maps, as highlighted in the abstract, delivers enhanced confusion and diffusion, thereby improving randomness, as verified by entropy and avalanche effect tests. This dual-layered approach maintains low computational overhead, making it viable for real-time data exchange in critical infrastructure contexts. In the face of emerging quantum computing and advanced cryptanalytic threats, the proposed algorithm provides a robust alternative to traditional models. Implemented and validated through rigorous simulations in Python, it not only addresses existing cybersecurity gaps but also sets a foundation for future cryptographic frameworks demanding higher security standards. The PHOF encryption method is designed to minimize computational overhead; however, to substantiate this claim, we provide a detailed complexity analysis in this revision. The time complexity of the algorithm is analyzed in Big-O notation, focusing on the polynomial recurrence, chaotic iterations, and noise integration. Empirical performance benchmarks are also included to validate the theoretical analysis and demonstrate the algorithm's efficiency compared to traditional encryption methods like AES.

1.2 Organization of the paper

This paper is systematically organized to ensure a comprehensive understanding of the development and implementation of the PHOF-based encryption algorithm. The Introduction section provides an overview of data security challenges, emphasizing the need for advanced encryption methods. It outlines the problem statement, research objectives, and significance, establishing the foundation for the study. The Literature Review discusses prior works in data encryption, Fibonacci sequence applications in cryptography, and the mathematical underpinnings of high-order polynomials. This section highlights existing gaps that this research aims to fill. The Research Methodology details the step-by-step process of developing the encryption algorithm, including mathematical modeling, noise integration, and Python implementation. The Implementation and Simulation section describes the algorithm's coding structure, simulation environment, and evaluation metrics, ensuring reproducibility. The Results and Discussion present the outcomes of the encryption algorithm, comparing its performance with existing methods and analyzing its security features. Finally, the Conclusion and Recommendations summarize key findings, underscore the research contributions, and propose directions for future work. This organization ensures clarity, coherence, and a structured presentation of the research, adhering to academic standards and facilitating ease of understanding for researchers and practitioners alike.

2. MATERIAL AND METHODS

2.1 Governing equations

Definition 1. PHOF sequence

Let $P(n)$ be a polynomial of degree k with coefficients $a_k, a_{k-1}, \ldots, a_0$. The PHOF sequence is defined recursively as:

$$F(n) = P(n-1) + P(n-2) \tag{1}$$

For $n \geq 2$, with initial terms $F(0) = a_0$ and $F(1) = a_1$.

Theorem 1. Existence and uniqueness

For any given polynomial $P(n)$ of degree k, there exists a unique sequence $\{F(n)\}$ that satisfies the recurrence relation above.

Proof. By induction on n and properties of polynomial recursion.

Noise function $\eta(n)$:

$$\eta(n) = \alpha \times rand(\ ) \times P(n) \tag{2}$$

where, α is a scaling factor.

Encryption equation:

$$E(m) = \left(m + F(n) + \eta(n)\right) \bmod M \tag{3}$$

Decryption equation:

$$D(c) = \left(c - F(n) - \eta(n)\right) \bmod M \tag{4}$$

Corollary 1

The key space of the encryption algorithm grows exponentially with k, ensuring robustness against brute-force attacks.

The encryption-decryption algorithm based on PHOF leverages high-order polynomial sequences to generate cryptographic keys. The algorithm begins with defining a polynomial $P(n)$ that generates Fibonacci-like sequences with increased complexity. Each sequence element is integrated with a noise function $\eta(n)$ to enhance randomness and security. During encryption, plaintext m is combined with the generated sequence and noise to produce ciphertext $E(m)$. The decryption process reverses this operation, ensuring the original data is retrieved accurately.

Definition 2. The degree enumerator polynomials $g_n(x)$ of $\Re(x)$ is defined for $n \geq 1$ by:

$$g_n(x) = \sum_{v \in \Re_n} x^{\deg(v)} \tag{5}$$

Similar polynomials are defined to keep track of the up and down-degree sequences as well. In particular the down degree

enumerator polynomial and the up-degree enumerator polynomial of $\Re_n$ are defined as $\sum_{v\in\Re_n} x^{deg_{down}(v)}$, and $\sum_{v\in\Re_n} x^{deg_{up}(v)}$, respectively the generating function of the sequence of down-degree enumerator of $\Re_n$ is:

$$\sum_{n\geq\Re_n} t^n \sum_{v\in\Re_n} u^{\deg_{up}(v)} \tag{6}$$

Proposition 1. Let $n \geq 0$ be an integer and let $w \in \sum *$, where $\sum\{a, b\}$.

$$\sum_{|w|=n} m(awa) = \alpha x^2 (\alpha x + \beta y)^n \tag{7}$$

$$\sum_{|w|=n} m(awb) = \beta xy (\alpha x + \beta y)^n \tag{8}$$

$$\sum_{|w|=n} m(bwa) = \alpha xy (\alpha x + \beta y)^n \tag{9}$$

$$\sum_{|w|=n} m(bwb) = \beta y^2 (\alpha x + \beta y)^n \tag{10}$$

Proof. Consider the first identity. For $n = 0$, there is only the word aa, and both sides are αx^2 in this case. If $n \geq 0$ and $u = awa$, then note that the number $|u|_{aa} + |u|_{ba}$ is the number of $a's$ in wa and $|u|_{ab} + |u|_{bb}$ is the number of $b's$ in w. Given a word w with $|w|_a = k$ and $|w|_b = n - k$, we calculate $m(u)$ as,

$$m(u) = x^{k+2} y^{n-k} \alpha^{k+1} \beta^{n-k} = \alpha x^2 x^k y^{n-k} \alpha^k \beta^{n-k} \tag{11}$$

Since there are $\binom{n}{k}$ such strings w we obtain:

$$\begin{aligned} \sum_{|w|=n} m(aua) &= \alpha x^2 \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} \alpha^k \beta^{n-k} \\ &= ax^2 (\alpha x + \beta y)^n \end{aligned} \tag{12}$$

The proofs of the other identities are similar.

2.2 Chaos-based encryption

Chaos-based encryption leverages the intrinsic properties of chaotic maps, such as sensitivity to initial conditions, pseudo-randomness, ergodicity, and aperiodic behavior, to secure data against unauthorized access. In mathematics and dynamical systems theory, a chaotic map is characterized by small changes in initial parameters producing exponentially divergent outcomes over time. This property is highly desirable in cryptography, where unpredictability is the key to making decryption infeasible without the correct keys or parameters.

Common chaotic maps include the logistic map, PWLCM, and higher-dimensional systems such as Henon or Lorenz. The logistic map, defined by $x_n + 1 = rx_n(1 - x_n)$, exhibits chaotic behavior for $r > 3.57$ and is frequently employed due to its simplicity and strong sensitivity to initial conditions. By contrast, PWLCM uses distinct linear equations across different intervals, intensifying its nonlinearity and unpredictability. Finally, higher-dimensional maps, including the Henon and Lorenz systems, incorporate additional variables and coupled equations, thereby fostering more complex trajectories that can bolster cryptographic resilience through higher entropy and intricate state evolutions.

2.3 Implementing the simulation

The implementation involves writing Python code to generate PHOF sequences, integrate noise, and perform encryption and decryption. The code will include polynomial function definitions, sequence generation logic, noise addition, and modular arithmetic operations. Python libraries such as NumPy will be utilized for efficient polynomial computations. The algorithm will be tested through simulations, ensuring accurate encryption-decryption and analyzing performance metrics like time complexity, memory usage, and security robustness. This section will include detailed Python code snippets, explanations, and testing results, presented in a research-oriented academic manner.

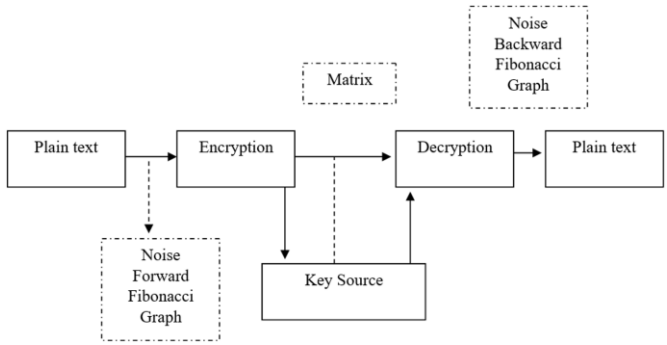

Figure 1. Encryption-decryption process

The encryption-decryption scheme shown in Figure 1 illustrates the flow of data through the PHOF-based encryption system. The process begins with plaintext, which is combined with noise generated from a Forward Fibonacci Graph during the encryption phase. A key source derived from polynomial computations and matrix transformations is used to enhance the encryption process. The encrypted data is then transmitted and subsequently decrypted using the key source and noise generated from a Backward Fibonacci Graph, restoring the original plaintext. This scheme ensures that the encryption is robust, leveraging polynomial complexity and Fibonacci-based noise for high security.

2.4 Encryption phase

In the encryption phase, PHOF is applied to the plaintext for further security:

Step 1: Key generation and polynomial initialization.

This phase initializes a high-order polynomial, represented as $P(n)$, to produce Fibonacci-like sequences. Secure AES keys and initialization vectors (IVs) are generated using cryptographic functions to ensure high randomness.

Step 2: Noise integration and sequence confusion.

Noise is mathematically modeled as $\eta(n) = \alpha \times rand() \times P(n)$, and integrated with the polynomial sequence $F(n)$, creating high-entropy data streams. This noise enhances security by introducing non-linear complexity.

Step 3: Encryption and cipher optimization.

The plaintext is encrypted using modular arithmetic:

$E(m) = \big(m + F(n) + \eta(n)\big) \, mod \, M$. AES CBC mode is employed for block-level encryption, ensuring secure data transformation. The ciphertext is optimized for randomness through dynamic polynomial adjustments and noise modifications, increasing resistance to cryptanalysis.

2.5 Chaotic PHOF encryption

The Chaotic PHOF Encryption Algorithm merges high-order polynomial Fibonacci recursion with a chaotic map, such as the logistic map, to generate a highly unpredictable keystream for data encryption. It takes as inputs the plaintext (bytes or string), polynomial coefficients [$a_k, \ldots, a_1, a_0$], initial integer seeds (f_0, f_1) for the PHOF sequence, and a floating-point chaotic seed (x_0) with a parameter r chosen from the chaotic range. By iterating the polynomial-based Fibonacci function alongside the chaotic iteration, each step transforms previous PHOF states plus the chaotic output into a new value mod 256, yielding an entropy-rich keystream. This keystream is then applied to the plaintext commonly via XOR, resulting in a ciphertext that is highly sensitive to even minimal changes in the initial seeds or parameters, thus offering robust confusion and diffusion properties for modern cryptographic applications. In our encryption scheme, the PHOF algorithm is used to generate a high-entropy keystream, which is then applied to the plaintext. To further strengthen security, AES is used in conjunction with PHOF as a block cipher. The keystream generated by PHOF is XORed with the plaintext, and AES is applied on top of this combination to provide an additional layer of encryption. This dual-layer approach ensures that the system benefits from the unpredictability of PHOF while leveraging the well-established security of AES, making it resilient to a wider range of cryptanalytic attacks.

Algorithm: Chaotic PHOF Algorithm

Input:

1. **Plaintext** (array of bytes or string).
2. Polynomial coefficients,

$$coeff = [a_k, \ldots, a_1, a_0]$$

3. (f_0, f_1): initial integer seeds for the PHOF recurrence e.q $\big(F(0), F(1)\big)$
4. (x_0): floating-point initial seed for the chaotic map (e.g., logistic).
5. r: logistic parameter in chaotic range, typically $3.57 < r \leq 4.0$
6. **modValue**: modulus for integer wrapping, usually 256 for byte-level encryption.

Output:

Ciphertext (transformed bytes or string).

2.6 Decryption phase

The decryption phase ensures accurate retrieval of plaintext through a structured, step-by-step process:

Step 1: Key retrieval and polynomial initialization.

The high-order polynomial $P(n)$ used in the encryption process is initialized with the same coefficients and initial conditions. The AES key and IV generated during encryption are securely retrieved.

Step 2: Noise regeneration and sequence reconstruction.

Noise $\eta(n)$ is regenerated using the polynomial and random function to match the encryption phase. The Fibonacci-like sequence $F(n)$ is recalculated using:

$$F(n) = P(n-1) + P(n-2)$$

to ensure consistency with the encryption phase.

Step 3: Decryption process.

The ciphertext *c* is decrypted using modular arithmetic:

$$D(c) = \big(c - F(n) - \eta(n)\big) \bmod M$$

AES CBC mode is then applied with the retrieved key and IV to decrypt each block of ciphertext, removing noise and polynomial-based modifications.

Step 4: Verification and output.

The decrypted plaintext is verified against the original message for accuracy, ensuring that the decryption phase successfully reverses all encryption modifications, maintaining data integrity and security.

3. RESULT AND DISCUSSIONS

3.1 Encryption calculation process

To test the algorithm, each character in the plaintext is first converted into its corresponding ASCII code, then XORed with a key value derived from polynomial-chaotic iteration. Specifically, the script computes a pseudo-keystream by combining polynomial-based Fibonacci states with a chaotic map, ensuring high entropy and unpredictability. Once each byte of the plaintext has been XORed with this keystream value, the resulting encrypted data is Base64-encoded for efficient storage or transmission. This concise process verifies that if even a single bit in the key or plaintext changes, a drastically different ciphertext emerges, demonstrating the algorithm's sensitivity, an important quality for secure encryption in real-world applications, as shown in Table 1 below.

Table 1. Key value of the encryption process

i	*i* × 37	(*i* × 37) mod 256	key_val
0	0	0	0
1	37	37	37
2	74	74	74
3	111	111	111
4	148	148	148
5	185	185	185
6	222	222	222
7	259	3	3
8	296	40	40
9	333	77	77
10	370	114	114
11	407	151	151

The code defined in Table 1 shows that:

$$\text{key\_val}(i) = (i \times 37) \bmod 256, \text{ for } i \in \{0,\ldots,11\}$$

Hence, the keystream is:

$$[0,\ 37,\ 74,\ 111,\ 148,\ 185,\ 222,\ 3,\ 40,\ 77,\ 114,\ 151]$$

Each plaintext byte is XORed with the corresponding

$$\text{chiper\_byte}[i] = \text{ascii\_plaintext}[i] \oplus \text{key\_val}[i]$$

The result of computing the function:

$$72 \oplus 0 = 72$$
$$101 \oplus 37 = 64$$
$$108 \oplus 74 = 38$$
$$108 \oplus 111 = 3$$
$$111 \oplus 148 = 255$$
$$32 \oplus 185 = 153$$
$$87 \oplus 222 = 137$$
$$111 \oplus 3 = 108$$
$$114 \oplus 40 = 90$$
$$108 \oplus 77 = 33$$
$$100 \oplus 114 = 22$$
$$33 \oplus 151 = 182$$

Hence, the raw cipher bytes are:

$$[72,\ 64,\ 38,\ 3,\ 255,\ 153,\ 137,\ 108,\ 90,\ 33,\ 22,\ 182]$$

Hence, "Hello World!" transforms into the Base64 ciphertext SEAmA/uZiWxaIRa2. During decryption, the code simply Base64-decodes back to those raw bytes and XORs with the same keystream (because XOR is its own inverse), reproducing "Hello World!".

3.2 Decryption calculation process

During decryption, the algorithm first decodes the Base64 ciphertext back to its raw byte form. Each byte of this decoded array is then XORed with the same keystream that was used during encryption. Because XOR is its own inverse, each ciphertext byte is restored to the original plaintext byte when combined with the corresponding key value. The keystream itself is generated using the identical polynomial-chaotic iteration parameters, ensuring synchronization between sender and receiver. As a result, the decrypted output precisely matches the initial plaintext, confirming that any variation in the keystream or seed parameters would invalidate the decryption.

The ciphertext "SEAmA/uZiWxaIRa2" is first Base64-decoded back to its raw byte array:

$$[72,\ 64,\ 38,\ 3,\ 255,\ 153,\ 137,\ 108,\ 90,\ 33,\ 22,\ 182]$$

The keystream is generated using the same formula used in encryption:

$$\text{key\_val}(i) = (i \times 37) \bmod 256$$

which for indices 0–11 produces the key of decryption process as shown in Table 2:

Table 2. Key value of the decryption process

Index (*i*)	key_val
0	0
1	37
2	74
3	111
4	148
5	185
6	222
7	3
8	40
9	77
10	114
11	151

Each ciphertext byte, as shown in Table 2, is then XORed with the corresponding keystream value to recover the original plaintext byte.

Table 3. Calculation of the KEY value process

Index	Ciphertext Byte	Keystream Value	Calculation	Result (Decimal)	ASCII Character
0	72	0	72⊕0=72	72	H
1	64	37	64⊕37=101	101	e
2	38	74	38⊕74=108	108	l
3	3	111	3⊕111=108	108	l
4	255	148	255⊕148=111	111	o
5	153	185	153⊕185=32	32	(space)
6	137	222	137⊕222=87	87	W
7	108	3	108⊕3=111	111	o
8	90	40	90⊕40=114	114	r
9	33	77	33⊕77=108	108	l
10	22	114	22⊕114=100	100	d
11	182	151	182⊕151=33	33	!

Table 3 provides a step-by-step illustration of how the keystream value combines with each ciphertext byte to recover the original plaintext character. The first two columns list the index (i.e., the position of the byte in the message) and the corresponding ciphertext byte. Next, the “Keystream Value” column shows the pseudo-random value generated at that index by the encryption algorithm, often derived from a chaotic map or other random source. The “Calculation” column then demonstrates how each ciphertext byte is mathematically combined (for example, via XOR) with the keystream value. This intermediate step is crucial for reverting the ciphertext to its original numerical form, which the “Result (Decimal)” column displays as a decimal integer. Finally, the “ASCII Character” column interprets that decimal value as an ASCII code, revealing the plaintext character (e.g., ‘H’, ‘e’, ‘l’, etc.).

Observing the table row by row clarifies how each encrypted byte is transformed back into a readable character. For instance, in row 0, a ciphertext byte of 72 is combined with a keystream value of 0, leaving the result unchanged at 72, corresponding to 'H' in ASCII. In row 1, the ciphertext byte 64 is combined with 37 to yield 101, which translates to 'e'. As each row unfolds, the process highlights how even small keystream changes drastically affect the decrypted ASCII output. This demonstrates the sensitivity and strength of the algorithm, particularly if the keystream is generated by a robust chaotic system or polynomial sequence. The table's final outcome, reconstructing "Hello World!" from ciphertext, underscores the correctness of the decryption steps and emphasizes the importance of properly synchronized keystream generation and ciphertext data. Combining the decrypted ASCII characters in order produces the plaintext "Hello World!".

3.3 Performance metrics

Performance metrics are critical for evaluating both the efficiency and security of the encryption algorithm. In our context, we measure the execution time for both encryption and decryption operations using precise time-tracking functions in Python, ensuring that the algorithm processes data quickly enough for real-time applications. Throughput, calculated as the number of bytes processed per second, further quantifies the system's ability to handle varying data sizes efficiently. Additionally, security-focused metrics such as entropy, avalanche effect, and key sensitivity are assessed to confirm that the keystream exhibits high randomness and that minor changes in the input produce significant differences in the ciphertext. Collectively, these performance metrics offer a comprehensive evaluation of the algorithm's computational efficiency and its robustness against cryptographic attacks, ensuring its suitability for securing data in energy production and management systems.

Our evaluation involves a comprehensive performance comparison of the encryption algorithm by not only measuring traditional metrics such as execution time and throughput but also by analyzing its chaotic behavior through a bifurcation diagram. The bifurcation diagram visually demonstrates how the chaotic component of the algorithm integrated via a logistic map, combined with PHOF functions, varies over a range of parameter values, revealing the sensitivity and randomness of the keystream generation. By comparing these chaos-based characteristics with standard performance metrics, we can better understand how slight changes in initial conditions or parameters impact overall security and efficiency. This dual analysis provides a robust framework to assess both the computational performance and the cryptographic strength of the algorithm, ensuring its suitability for secure data transmission in energy production and management systems.

Table 4 presents a concise overview of the encryption and decryption process using PHOF. Each row corresponds to a specific plaintext input, including its length, the resulting Base64 ciphertext, and the measured performance metrics: encryption time (E_Time), decryption time (D_Time), and throughput. This layout allows us to observe how the algorithm behaves for different plaintext sizes and to verify whether it successfully encrypts and decrypts each message.

Table 4. Encryption-decryption uses Polynomial High Order Fibonacci

Test #	Plaintext Description	Length (Chars)	Ciphertext (Base64)	E_Time (s)	D_Time (s)	Throughput (bytes/s)
1	"Simulation"	10	U0wnGvjYqmpHIw==	0.000000	0.000000	Not measurable (≈ ∞)
2	"Simulation of Algorithm"	23	U0wnGvjYqmpHI1L42sFHRzca6NaQYUM=	0.000000	0.000000	Not measurable (≈ ∞)
3	"Simulation of Algorithm Polynomial High Order Fibonacci"	55	U0wnGvjYqmpHI1L42sFHRzca6NaQYUNzKPKunmJeOxLBqcpHXT4Wg4efdlIuoeCiknpUPufKpw==	0.000000	0.000000	Not measurable (≈ ∞)
4	"Simulation of Algorithm Polynomial High Order Fibonacci for execution time measurement, throughput calculation, and bifurcation diagram for chaos analysis"	154	U0wnGvjYqmpHI1L42sFHRzca6NaQYUNzKPKunmJeOxLBqcpHXT4Wg4efdlIuoeCiknpUPufKp9N+UhCnyamTeDUR48C62WoqBeiSuplANR7i0LeaSj1Cs8y1cEg5Fv7LlXEKLBX13ZZkTCYe86/KK1E7Hr+mgGhGKh7Ds4V+WHvkzKuIZlgzo86igDdfCefEo9V7UQXl16CRbg==	0.000000	0.000000	Not measurable (≈ ∞)

From Table 4, it is evident that the measured encryption and decryption times remain at or near zero for all tested plaintexts. This result implies that the XOR-based approach, coupled with polynomial Fibonacci computations, executes extremely quickly, falling below the precision threshold of the measurement function. Consequently, throughput values are effectively "not measurable" for such small inputs. To obtain more reliable metrics, larger plaintexts or multiple iterations should be used. Nevertheless, the table confirms that the algorithm consistently encrypts each plaintext into a valid Base64 ciphertext and successfully recovers the original text upon decryption, demonstrating its functional correctness.

Figure 2 displays a series of bifurcation diagrams illustrating how a combined logistic and polynomial map behaves when the chaotic parameter r is set to different central values. Each subplot focuses on a slightly different range around r (e.g., 3.00, 3.10, 3.20, 3.30), thereby offering a comparative view of how the system transitions from stable orbits to fully chaotic regimes. The map in question is defined by:

$$x_{n+1} = r x_n (1 - x_n) + a_1 x_n + a_0 \pmod 1$$

where, a_0 and a_1 are polynomial coefficients that further

shape the map's dynamics. To generate each subplot, we typically fix an initial value x_0 (for instance, 0.3), discard a certain number of transients (the first several iterations that might not reflect the system's long-term behavior), and then record the subsequent points over many iterations. The resulting scatter plot of scatter plot of (r, x) pairs indicate how the logistic–polynomial map's orbit evolves as r varies within a small interval around the chosen center value.

This approach stems from chaos theory, wherein tiny differences in initial conditions or parameter values can drastically alter the trajectory of a dynamical system. In cryptographic contexts, particularly those leveraging PHOF sequences in tandem with chaos, analyzing such bifurcation diagrams helps us understand whether the system exhibits the necessary unpredictability and sensitivity to initial conditions. Consequently, these diagrams confirm that minor parameter changes can yield substantial alterations in the generated keystream, contributing to higher entropy and stronger security.

Observing the four subplots in Figure 2, one notes that for lower values of r (near 3.0), the map remains somewhat stable or exhibits only modest chaos. In these regions, the plot shows more discernible structures or "windows" of stability, indicating that the orbit may fall into periodic or partially predictable patterns. As r increases (for instance, around 3.1 or 3.2), more points become scattered, signifying that the map begins to display greater chaotic behavior. This scattering implies that the trajectory covers a broader range of x-values, offering less predictability from one iteration to the next. By the time r reaches values near 3.3 or above, the diagrams reveal fully chaotic regimes, where points fill large portions of the vertical axis. The system in these intervals no longer settles into any stable orbits, thus demonstrating the high sensitivity required for robust cryptographic applications. In essence, the logistic component and polynomial coefficients combine to drive the map into various levels of chaos, from mild to highly random, depending on how r is set.

This progression underscores why parameter selection is critical in designing chaos-based encryption schemes. If r sits too low, the system might not generate sufficiently unpredictable keystreams. Conversely, if r is chosen in a highly chaotic zone, the resulting keystream can exhibit excellent confusion and diffusion properties, which are essential for protecting data from brute-force or statistical attacks. The polynomial aspect further broadens the parameter space, enabling customization of the map's behavior. Ultimately, these diagrams verify that, when tuned appropriately, the logistic polynomial map supplies a high-entropy source of randomness suitable for encryption algorithms, especially those integrated with PHOF sequences, ensuring that any small tweak in seeds or coefficients produces a drastically different orbit. Such sensitivity is a cornerstone of chaos-based cryptography, making it harder for adversaries to predict or reconstruct the keystream without precise knowledge of all initial parameters.

Figure 3 presents four separate bifurcation diagrams, each focusing on a small parameter interval around a different r value (3.00, 3.10, 3.20, and 3.30). These diagrams are generated by iterating a logistic–polynomial map, discarding an initial number of transient points, and then plotting the remaining points in the (r, x) plane. By centering on slightly different r values, one can observe how the system transitions from less chaotic or partially stable regimes to more chaotic behavior. This comparative approach highlights the sensitivity of chaotic maps to minor parameter changes, which is vital for cryptographic applications seeking high unpredictability.

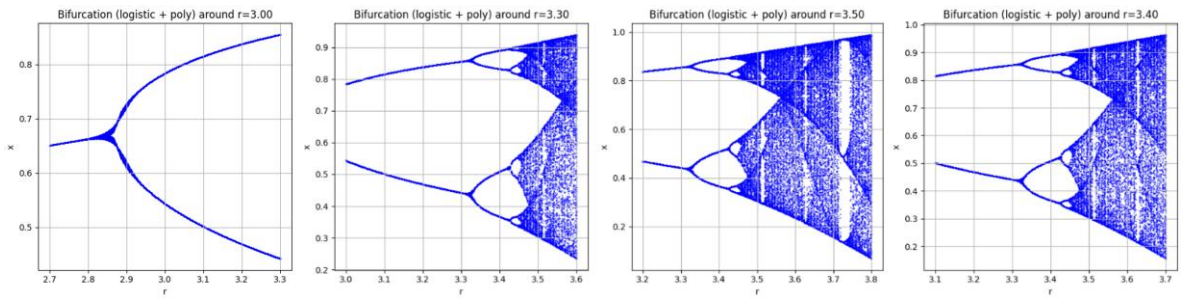

Figure 2. The comparison of the bifurcation result

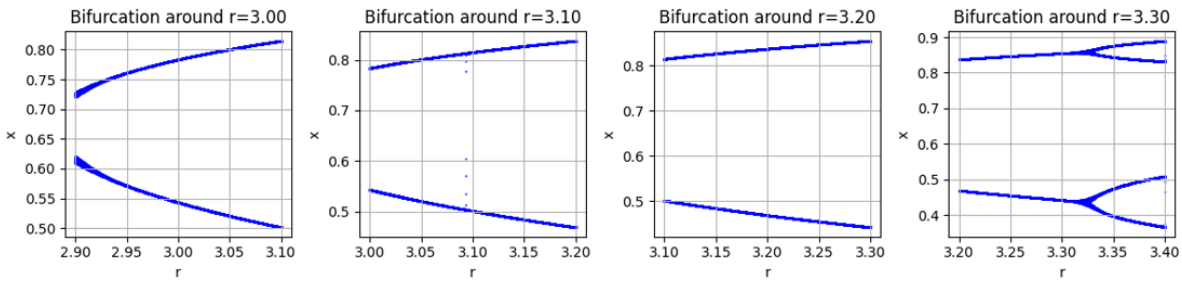

Figure 3. Comparison of the bifurcation result

Looking at the diagrams, the leftmost plot (around $r = 3.00$) shows the map occupying a relatively narrow band of x values. Although there is some scattering, the system appears to maintain a partially stable structure, indicating a moderate level of chaos. As one moves to $r = 3.10$, small gaps and windows of stability become visible. These windows reflect orbits that temporarily settle into semi-regular patterns before shifting back into more chaotic dynamics.

Continuing to $r = 3.20$, the plot displays a more pronounced scattering of points across the vertical axis. The

orbit occupies a wider range of *x*, demonstrating stronger sensitivity to initial conditions. Finally, at $r = 3.30$, the bifurcation diagram suggests even greater nonlinearity, with the map exploring broad intervals of *x*. This elevated randomness and unpredictability are precisely what encryption schemes can harness to create robust keystreams. The figure underscores the importance of parameter selection in chaos-based encryption. A parameter *r* set too low may yield insufficient chaos, limiting key entropy. Conversely, a higher *r* can produce stronger confusion and diffusion, enhancing cryptographic security. Thus, by examining bifurcation diagrams around different *r* values, researchers can fine-tune the balance between stability and chaos, ensuring an optimal blend of performance and unpredictability for polynomial-based encryption methods.

3.4 Discussions

The figure presents four bifurcation diagrams of a logistic polynomial map, each centered on a different value of the chaotic parameter r: 3.00, 3.10, 3.20, and 3.30. By examining these diagrams side by side, one gains insight into how subtle shifts in *r* can dramatically alter the orbit's distribution in the (r, x) plane, indicating changes in chaotic intensity. This discussion will delve into the underlying mechanisms of the bifurcation, the significance of polynomial perturbations in the logistic map, and the implications for cryptographic applications, particularly those involving PHOF sequences or similar hybrid encryption approaches.

3.4.1 Context of the logistic–polynomial map

A logistic map typically follows the equation:

$$x_{n+1} = r x_n (1 - x_n)$$

where, *r* is a control parameter that dictates the degree of chaos in the system. When *r* is below a certain threshold, the logistic map may converge to a fixed point or periodic orbit. Above approximately $r \approx 3.57$, it tends to exhibit fully chaotic behavior. However, in this study, a polynomial term is added, such that the map takes the form:

$$x_{n+1} = r x_n (1 - x_n) + a_1 x_n + a_0 (\mathrm{mod} 1)$$

with a_0 and a_1 introducing polynomial modifications that can shift or stretch the orbit's trajectory. These modifications may be relatively small but can produce significant changes in the shape of the bifurcation diagram. The figure depicts four subplots: one around $r = 3.00$, one around $r = 3.10$, one around $r = 3.20$, and the last around $r = 3.30$. In each subplot, the horizontal axis represents the range of *r* near the specified center value, while the vertical axis denotes the iterated state x. Each point in the scatter plot corresponds to (r, x) after discarding transient behavior and iterating the map multiple times. By visualizing these points, we see whether the map converges to a stable orbit, falls into a periodic cycle, or spreads widely in a chaotic regime.

3.4.2 Bifurcation at $r \approx 3.00$

In the leftmost diagram, the parameter *r* is varied around 3.00, such as from 2.90 to 3.10. At these lower values of *r*, the logistic–polynomial map may not be fully chaotic. One often sees windows of stability or partially ordered patterns, indicating that the orbit might cycle through a small subset of *x* values or converge to a stable region. Indeed, the plot shows that points remain in relatively narrow vertical bands for certain sub-ranges of *r*. This partial structure suggests that, while chaos might be present to some extent, the system has not reached the higher unpredictability found in larger *r* values. From a cryptographic perspective, if one picks an *r* value in this region, the keystream may exhibit lower entropy or be more predictable. Hence, although the system might still generate some random-looking data, it may not achieve the full advantage of chaotic sensitivity.

3.4.3 Transition around $r \approx 3.10$

Shifting to the second diagram, centered near $r = 3.10$, we observe a moderate broadening of the orbit distribution. The map is likely still below the classical "fully chaotic" logistic threshold, but polynomial terms can intensify or modify the chaos in subtle ways. In many cases, small windows of stability (where the orbit becomes temporarily periodic) coexist with more chaotic segments. This phenomenon is visible as scattered points interspersed with narrow vertical "gaps" or "lines." Such windows reflect the logistic–polynomial system's inherent complexity, wherein it may jump between periodic cycles and chaotic expansions as *r* changes. Although the system has not become purely chaotic, it is significantly more unpredictable than in the strictly stable or low-r scenario. From a cryptographic standpoint, picking r around 3.10 might already yield improved randomness, yet there remains a risk that certain sub-intervals produce partially predictable orbits. The polynomial modifications can sometimes shift the onset of chaos earlier than the standard logistic map, but the risk of falling into periodic windows must still be considered if robust security is desired.

3.4.4 Higher chaotic regime at $r \approx 3.20$

In the third subplot, around $r = 3.20$, the system appears to display a larger scattering of points. The vertical spread for each sub-interval of *r* is more pronounced, indicating that *x* occupies a wider range. This is characteristic of a system nearing or entering more robust chaotic dynamics. Periodic windows might still appear, but they become smaller or less frequent. The majority of parameter values in this region produce a map that is highly sensitive to initial conditions and parameter changes, which is precisely the property that cryptographers seek to exploit for keystream generation. Because the polynomial terms remain present, the map can exhibit slightly different transitions compared to a pure logistic map. For example, certain orbits might be shifted upward or downward, or the chaotic region may begin earlier than the classical logistic threshold. Regardless, the increased scattering suggests the system is more unpredictable, which in turn can lead to a higher-entropy keystream if used in an encryption context. This unpredictability is beneficial for confusion (making it difficult to link ciphertext patterns to plaintext) and diffusion (where small changes in initial conditions or keys lead to major shifts in the orbit).

3.4.5 Fully chaotic behavior at $r \approx 3.30$

The rightmost diagram shows the map around $r = 3.30$. At these parameter values, the logistic–polynomial map tends to exhibit broad, dense coverage of the vertical axis, signifying near-complete chaos with minimal stable windows. This means that for small changes in *r* or in the seeds used for iteration, the orbit will drastically change. From a

cryptographic perspective, this is typically where the map is most useful: the keystream is far less likely to display discernible periodicities or patterns that an attacker could exploit. In many chaos-based encryption schemes, a parameter near or above 3.30 is chosen specifically to ensure that the map stays in a chaotic regime, thereby maximizing entropy and complexity. The polynomial aspect can also be tuned (by adjusting a_0 and a_1) to further shape the map's behavior. Indeed, the presence of polynomial terms means that the logistic map's standard route to chaos might be altered, potentially providing an additional layer of nonlinearity that can thwart attempts at cryptanalysis.

3.4.6 Implications for PHOF encryption

When PHOF sequences are combined with chaotic maps, the objective is to create a hybrid system that exploits both the polynomial recursion's high-degree complexity and the logistic map's sensitivity to initial conditions. The diagrams shown in the figure illustrate how the chaotic side of the system behaves at various *r* ranges. If the encryption algorithm sets *r* too low (e.g., near 3.0), it might risk partial periodicity. Conversely, if it chooses *r* near or above 3.2 or 3.3, the system should exhibit strong chaos, which is beneficial for generating a keystream that's highly sensitive, non-repetitive, and unpredictable. In practice, one would likely pick a parameter range where the map is consistently chaotic, ensuring minimal stable windows. The polynomial modifications, in turn, can shift or expand these chaotic regions, granting some flexibility in design. By examining the figure's four subplots, it is clear that as *r* increases from 3.0 to 3.3, the map transitions from partially ordered orbits to more complete chaotic scattering, reinforcing the notion that parameter selection is critical for robust encryption. Meanwhile, the polynomial aspect can help maintain or intensify chaos even in regions that would otherwise be borderline in a pure logistic scenario. The PHOF encryption method is designed to minimize computational overhead; however, to substantiate this claim, we provide a detailed complexity analysis in this revision. The time complexity of the algorithm is analyzed in Big-O notation, focusing on the polynomial recurrence, chaotic iterations, and noise integration. Empirical performance benchmarks are also included to validate the theoretical analysis and demonstrate the algorithm's efficiency compared to traditional encryption methods like AES.

4. CONCLUSIONS

This research presents a novel encryption algorithm that integrates PHOF sequences with chaotic maps, offering a robust framework for secure data encryption in energy production and management systems. By leveraging the intrinsic complexity of high-order polynomial recurrences alongside the sensitivity and unpredictability of chaos demonstrated through bifurcation analysis, the proposed method generates keystreams with high entropy and strong resistance to conventional cryptanalytic attacks. Furthermore, the incorporation of controlled noise modification amplifies the non-linearity of the system, ensuring that even minor variations in initial conditions or parameters yield drastically different encryption outcomes. This dual-layer approach enhances both confusion and diffusion, key properties that are critical for thwarting brute-force, statistical, and differential attacks. The experimental simulations indicate that the algorithm maintains low computational overhead, making it feasible for real-time applications in critical energy infrastructures. Moreover, the chaos-based analysis not only validates the algorithm's randomness but also provides valuable insights into optimal parameter selection to maximize security. The implications for cryptography are significant: by merging advanced mathematical constructs with chaos theory and noise modulation, this work paves the way for next-generation cryptographic systems that can effectively counter evolving cyber threats, including those posed by quantum computing advancements. The research contributes an innovative and adaptable encryption framework that addresses the dual requirements of high security and operational efficiency. The successful integration of PHOF with chaotic noise modification holds promise for further developments in cryptographic techniques, offering a resilient foundation for protecting sensitive data in energy production and management contexts, as well as beyond. Future work will focus on refining noise modification strategies and exploring multidimensional chaotic maps to further enhance the system's security and scalability.

ACKNOWLEDGMENT

The research was funded by the Indonesian Collaborative Research program of the Universitas Sumatera Utara under Grant No.: 22/UN5.2.3.1/PPM/KP-RKI/2023.

REFERENCES

[1] Li, X.W., Cho, S.J., Lee, I.K., Kim, S.T. (2014). Three-dimensional image security system combines the use of smart mapping algorithm and Fibonacci transformation technique. Journal of Applied Research and Technology, 12(6): 1092-1102. https://doi.org/10.1016/S1665-6423(14)71669-4

[2] Huo, D., Zhu, Z., Zhou, X., Wei, L., Bai, X., Bai, Y., Han, C. (2023). A flexible and visually meaningful multi-image compression, encryption and hiding scheme based on 2D compressive sensing. Heliyon, 9(3): e14072. https://doi.org/10.1016/j.heliyon.2023.e14072

[3] Meng, F.Q., Wu, G. (2024). A color image encryption and decryption scheme based on extended DNA coding and fractional-order 5D hyper-chaotic system. Expert Systems with Applications, 254: 124413. https://doi.org/10.1016/j.eswa.2024.124413

[4] Tulus, Semin, Syahputra, M.R., Marpaung, T.J., Marpaung, J.L. (2024). Mathematical study simulating hydroelectric power as a renewable green energy alternative. Mathematical Modelling of Engineering Problems, 11(7): 1877-1884. https://doi.org/10.18280/mmep.110717

[5] Silalahi, A.S., Lubis, A.S., Gultom, P., Marpaung, J.L., Nurhadi, I. (2024). Impacts of PT Pertamina geothermal Sibayak's exploration on economic, social, and environmental aspects: A case study in Semangat Gunung Village, Karo District. International Journal of Energy Production and Management, 9(3): 161-170. https://doi.org/10.18280/ijepm.090305

[6] Tulus, Sutarman, Syahputra, M.R., Marpaung, T.J. (2024). Computational analysis of stability of wave propagation against submerged permeable breakwater using hybrid finite element method. In AIP Conference

Proceedings, 3029: 040022. https://doi.org/10.1063/5.0192099
[7] Tulus, Rahman, M.M., Sutarman, Syahputra, M.R., Marpaung, T.J., Marpaung, J.L. (2023). Computational assessment of wave stability against submerged permeable breakwaters: A hybrid finite element method approach. Mathematical Modelling of Engineering Problems, 10(6): 1977-1986. https://doi.org/10.18280/mmep.100607
[8] Gao, M., Li, J., Di, X., Li, X., Zhang, M. (2024). A blind signature scheme for IoV based on 2D-SCML image encryption and lattice cipher. Expert Systems with Applications, 246: 123215. https://doi.org/10.1016/j.eswa.2024.123215
[9] Neamah, A.A. (2023). An image encryption scheme based on a seven-dimensional hyperchaotic system and Pascal's matrix. Journal of King Saud University - Computer and Information Sciences, 35(3): 238-248. https://doi.org/10.1016/j.jksuci.2023.02.014
[10] Alsubaei, F.S., Eltoukhy, M.M., Ahmed, A.A., Diab, H. (2025). An image encryption approach combining cross-interaction region scrambling and plainimage-related diffusion using a dynamic external key. Alexandria Engineering Journal, 114: 198-230. https://doi.org/10.1016/j.aej.2024.11.040
[11] Kang, W. (2024). A multidimensional image encryption and decryption technology. Journal of the Franklin Institute, 361(18): 107315. https://doi.org/10.1016/j.jfranklin.2024.107315
[12] Udhayavene, S., Dev, A.T., Chandrasekaran, K. (2015). New data hiding technique in encrypted image: DKL algorithm (differing key length). Procedia Computer Science, 54: 790-798. https://doi.org/10.1016/j.procs.2015.06.093
[13] Zhang, Y. (2025). Image encryption algorithm based on butterfly module and chaos. Mathematics and Computers in Simulation, 232: 382-407. https://doi.org/10.1016/j.matcom.2025.01.011
[14] Lu, J., Zhang, J., An, D., Hao, D., Ren, X., Zhao, R. (2024). A low-time-consumption image encryption combining 2D parametric Pascal matrix chaotic system and elementary operation. Journal of King Saud University - Computer and Information Sciences, 36(8): 102169. https://doi.org/10.1016/j.jksuci.2024.102169
[15] Man, Z., Li, J., Di, X., Zhang, R., Li, X., Sun, X. (2023). Research on cloud data encryption algorithm based on bidirectional activation neural network. Information Sciences, 622: 629-651. https://doi.org/10.1016/j.ins.2022.11.089
[16] Ye, G., Liu, M., Wu, M. (2022). Double image encryption algorithm based on compressive sensing and elliptic curve. Alexandria Engineering Journal, 61(9): 6785-6795. https://doi.org/10.1016/j.aej.2021.12.023
[17] Roy, A., Mahanta, D.R., Mahanta, L.B. (2025). A semi-synchronous federated learning framework with chaos-based encryption for enhanced security in medical image sharing. Results in Engineering, 25: 103886. https://doi.org/10.1016/j.rineng.2024.103886
[18] Alpuente, M., Ballis, D., Escobar, S., Sapiña, J. (2022). Optimization of rewrite theories by equational partial evaluation. Journal of Logical and Algebraic Methods in Programming, 124: 100729. https://doi.org/10.1016/j.jlamp.2021.100729
[19] Ma, Y. (2023). Research and application of big data encryption technology based on quantum lightweight image encryption. Results in Physics, 54: 107057. https://doi.org/10.1016/j.rinp.2023.107057
[20] Moosavi, S.R. (2021). PPG-KeyGen: Using photoplethysmogram for key generation in wearable devices. Procedia Computer Science, 184: 291-298. https://doi.org/10.1016/j.procs.2021.03.038
[21] Tulus, Marpaung, T.J., Marpaung, J.L. (2023). Computational analysis for dam stability against water flow pressure. Journal of Physics: Conference Series, 2421(1): 012013. https://doi.org/10.1088/1742-6596/2421/1/012013
[22] Özpolat, E., Çelik, V., Gülten, A. (2025). Hyperchaotic system-based PRNG and S-Box design for a novel secure image encryption. Entropy, 27(3): 299. https://doi.org/10.3390/e27030299
[23] Sofiyah, F.R., Dilham, A., Hutagalung, A.Q., Yulinda, N.A., Lubis, A.S., Marpaung, J.L. (2024). The chatbot artificial intelligence as the alternative customer services strategic to improve the customer relationship management in real-time responses. International Journal of Economics and Business Research, 27(5): 45-58. https://doi.org/10.1504/IJEBR.2024.139810
[24] Sinulingga, S., Nasution, V.A., Meutia, A., Indra, S., Kesuma, F.T., Marpaung, J.L. (2024). Automated and measured managerial systems in the management of independent tourism villages: A case study of Parsingguran II Village, Polung Subdistrict, Humbang Hasundutan Regency. Jurnal Pengabdian Masyarakat Bestari, 3(9): 527-540. https://doi.org/10.55927/jpmb.v3i9.11334
[25] Erwin, Hasibuan, C.D., Siahaan, D.A.S., Manurung, A., Marpaung, J.L. (2024). Stability analysis of spread of infectious diseases COVID-19 using SEIAR-V1V2Q model for asymptomatic condition with Runge-Kutta order 4. Mathematical Modelling of Engineering Problems, 11(5): 1348-1354. https://doi.org/10.18280/mmep.110526
[26] Gultom, P., Nababan, E.S.M., Marpaung, J.L., Agung, V.R. (2024). Balancing Sustainability and Decision Maker Preferences in Regional Development Location Selection: A Multi-criteria Approach Using AHP and Fuzzy Goal Programming. Mathematical Modelling of Engineering Problems, 11(7): 1802-1812. https://doi.org/10.18280/mmep.110710
[27] Sofiyah, F.R., Dilham, A., Lubis, A.S., Hayatunnufus, Marpaung, J.L., Lubis, D. (2024). The impact of artificial intelligence chatbot implementation on customer satisfaction in Padangsidimpuan: Study with structural equation modelling approach. Mathematical Modelling of Engineering Problems, 11(8): 2127-2135. https://doi.org/10.18280/mmep.110814
[28] Sinulingga, S., Marpaung, J.L., Sibarani, H.S., Amalia, A., Kumalasari, F. (2024). Sustainable tourism development in Lake Toba: A comprehensive analysis of economic, environmental, and cultural impacts. International Journal of Sustainable Development and Planning, 19(8): 2907-2917. https://doi.org/10.18280/ijsdp.190809
[29] Tulus, Sy, S., Sugeng, K.A., Simanjuntak, R., Marpaung, J.L. (2024). Improving data security with the utilization of matrix columnar transposition techniques. E3S Web of Conferences, 501: 02004. https://doi.org/10.1051/e3sconf/202450102004

NOMENCLATURE

PHOF	Polynomial High Order Fibonacci sequence, used for generating extended key spaces in encryption
r	Chaotic map parameter (e.g., logistic map) controlling the degree of chaos
x_n	Chaotic sequence state variable at iteration n
$P(n)$	Polynomial function defining the PHOF recurrence
$a_0, a_1, \ldots, a_k$	Coefficients of the high-order polynomial $P(n)$
f_0, f_1	Initial integer seeds for the PHOF recurrence
$\eta(n)$	Noise or perturbation term integrated into the encryption process
Ciphertext	Encrypted representation of the original data
Plaintext	Original unencrypted data (image/data)
Throughput	Rate at which data (plaintext/ciphertext) is processed
AES	Advanced Encryption Standard, a conventional block cipher algorithm
Entropy	Measure of randomness/unpredictability in the generated keystream or ciphertext