Journal of Applied Data Sciences
Vol. 7, No. 3, September 2026, pp. 2434-2445

ISSN 2723-6471
2334

# A Hybrid AES–NTRU Cryptographic Framework with Deterministic Dynamic S-Box for Secure Data Transmission

Purwanto Simamora[1], Poltak Sihombing[2,*], Mohammad Andri Budiman[3], Jos Timanta Tarigan[4]

[1]*Postgraduate Program in Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Street Dr. T. Mansur No. 9, Medan 20155, Indonesia*

[2,3,4]*Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Street Dr. T. Mansur No. 9, Medan 20155, Indonesia*

(Received: February 1, 2026; Revised: April 5, 2026; Accepted: July 5, 2026; Available online: August 18, 2026)

**Abstract**

The rapid expansion of interconnected sensing and monitoring systems has significantly increased the volume of digital data exchanged across distributed environments. While these systems provide substantial benefits for real-time monitoring and automated decision making, they also introduce critical challenges related to data confidentiality and secure communication. This study proposes a hybrid cryptographic framework designed to strengthen secure data transmission in distributed sensing environments. The primary objective of this research is to integrate a symmetric encryption mechanism with a lattice-based public-key cryptographic scheme within a hybrid key protection architecture. In addition, a deterministic round-dependent substitution mechanism is introduced into the symmetric encryption process to provide controlled structural variability during data encryption while preserving compatibility with the standard round-based encryption structure. The proposed framework was implemented in a software-based prototype environment and evaluated through a series of encryption and decryption experiments using representative data payloads that emulate typical communication scenarios in distributed sensing systems. The evaluation focused on three aspects: implementation correctness, computational performance, and diffusion characteristics of the encryption process. Experimental results confirm that all encrypted data payloads were successfully decrypted to their original plaintext without bit errors, demonstrating the functional correctness of the integrated cryptographic architecture. The encryption process achieved an avalanche effect of 47.01 percent, with an average encryption time of 118.99 milliseconds and a decryption time of 5.91 milliseconds in the prototype environment. These results indicate consistent diffusion behavior within acceptable experimental bounds while maintaining reasonable computational overhead for gateway-level communication nodes. The findings suggest that integrating adaptive substitution mechanisms within hybrid cryptographic architectures can provide flexible encryption structures suitable for secure data transmission in distributed monitoring systems, while remaining compatible with widely deployed symmetric encryption frameworks.

*Keywords:* Hybrid encryption, AES–NTRU, cryptography, dynamic S-box, IoT security

## 1. Introduction

The growth of the Internet of Things (IoT) has not only resulted in a rapid increase in data exchange but has also introduced significant challenges related to data security and confidentiality [1], [2]. Applications such as environmental monitoring, industrial automation, smart infrastructure, and healthcare monitoring systems rely heavily on continuous data transmission, making them particularly vulnerable to security risks. To protect data confidentiality in such environments, various symmetric encryption algorithms, including the Advanced Encryption Standard (AES), have been widely implemented to secure IoT payload data [3], [4], [5].

AES is widely recognized as a secure and well-established symmetric encryption algorithm that has undergone extensive cryptographic analysis since its standardization. The reuse of a fixed substitution box (S-box) across encryption rounds is an inherent design characteristic of AES and does not constitute a vulnerability in the standard cryptographic model, as the substitution layer has been carefully designed to resist known attacks such as differential and linear cryptanalysis. However, in long-term IoT deployments where devices continuously encrypt large volumes

***Corresponding author: Poltak Sihombing (poltak@usu.ac.id)**
**DOI: https://doi.org/10.47738/jads.v7i3.1307**

of data under constrained computational environments, researchers have explored adaptive internal transformations to study their potential effects on empirical diffusion behavior and implementation-level diversity. In this context, adaptive substitution mechanisms have been investigated as a means of introducing controlled variability within the encryption process while preserving compatibility with the original AES structure. In this study, adaptive substitution behavior is explored primarily from an implementation and diffusion perspective rather than as a modification of the core cryptographic security assumptions of AES.

Recent advances in computational capabilities have also raised concerns regarding the long-term security of classical public-key cryptographic algorithms such as RSA. While RSA remains secure against known classical cryptanalytic attacks when sufficiently large key sizes are used, its security fundamentally relies on the hardness of integer factorization. This assumption becomes vulnerable in the presence of large-scale quantum computers, as Shor's algorithm can solve the integer factorization problem in polynomial time. Consequently, the emergence of quantum computing has motivated increasing research into post-quantum cryptographic schemes. Lattice-based cryptography, including the NTRU cryptosystem, has received considerable attention due to its strong security foundations and resistance to both classical and quantum attacks, making it a promising candidate for secure key encapsulation in future communication systems [6]. Furthermore, to balance performance efficiency and key protection, the Key Encapsulation Mechanism–Data Encapsulation Mechanism (KEM–DEM) paradigm has been widely adopted in modern hybrid cryptographic systems [7], [8]. In this architecture, symmetric ciphers are used for bulk data encryption while asymmetric mechanisms protect session keys. Recent studies have also explored adaptive behavior in symmetric encryption systems, particularly for long-term IoT deployments. Several approaches investigate modifications of cryptographic primitives by introducing adaptive transformations within existing encryption structures.

For example, research has explored deterministic substitution mechanisms applied to the AES SubBytes transformation while maintaining compatibility with the original round structure [9], [10]. In addition, several hybrid encryption frameworks have been proposed to enhance security in IoT environments. Ehsan investigates hybrid schemes that employ post-quantum Key Encapsulation Mechanisms such as Kyber and NTRU combined with AES as the data encapsulation mechanism, evaluating latency and robustness in IoT contexts [11]. Similarly, Ramakrishna proposes a hybrid post-quantum encryption framework (PSESV) integrating AES with post-quantum KEMs and demonstrates improved throughput performance in embedded environments [12]. However, despite these developments, several research gaps remain. First, most existing hybrid encryption frameworks focus primarily on improving key distribution security using post-quantum algorithms while leaving the internal structure of the symmetric encryption component unchanged. As a result, limited attention has been given to exploring adaptive internal transformations within symmetric ciphers that may influence empirical diffusion behavior and structural variability in long-term IoT deployments. Second, studies investigating modifications to AES often introduce significant structural changes to the algorithm, which may reduce compatibility with the standard AES round structure and complicate practical implementation. Consequently, there remains a need for approaches that introduce adaptive substitution mechanisms within AES while preserving the original algorithmic structure. To address these gaps, this study proposes a hybrid AES–NTRU encryption model integrated with a deterministic round-dependent dynamic S-box mechanism. The proposed approach aims to introduce controlled variability within the AES substitution stage while maintaining full compatibility with the standard AES round structure. The encryption framework is implemented within a hybrid architecture following the KEM–DEM paradigm, where NTRU is used for session-key encapsulation and AES is used for payload encryption. In addition, the proposed model incorporates Pretty Good Privacy (PGP)-based key validation to ensure the authenticity of session keys during communication.

The contributions of this study can be summarized as follows. First, this research designs a hybrid AES–NTRU encryption model that follows the Key Encapsulation Mechanism–Data Encapsulation Mechanism (KEM–DEM) paradigm to support secure data transmission in Internet of Things (IoT) environments [13]. Second, the study introduces a deterministic dynamic S-box into the AES encryption process while preserving the standard AES round structure, enabling controlled variability in the substitution stage without modifying the fundamental algorithmic architecture [14], [15]. Third, the proposed model is experimentally evaluated using representative IoT data payloads to analyze its computational performance and empirical diffusion characteristics, providing insights into the practical feasibility of the hybrid encryption framework in gateway-level IoT communication systems [16], [17].

## 2. Literature Review

### 2.1. Cryptography for IoT Systems

The emergence of advanced computational models poses long-term challenges to conventional public-key cryptographic algorithms such as RSA and elliptic curve cryptography. Consequently, increasing attention has been directed toward alternative cryptographic approaches capable of providing stronger security guarantees for Internet of Things (IoT) communication systems. Among these approaches, lattice-based cryptography has received significant interest due to its strong security foundations and practical implementations. The NTRU cryptosystem, in particular, has been widely studied as an efficient key encapsulation mechanism suitable for embedded and IoT platforms. Experimental evaluations show that NTRU supports efficient polynomial arithmetic operations while maintaining competitive performance on constrained devices, making it a promising candidate for protecting cryptographic keys in IoT communication systems [18], [19], [20]. In secure communication architectures, hybrid cryptographic systems are commonly employed to combine the efficiency of symmetric encryption with the key protection capabilities of asymmetric cryptography. Many implementations follow the Key Encapsulation Mechanism–Data Encapsulation Mechanism (KEM–DEM) paradigm, where symmetric encryption protects bulk data while asymmetric cryptography secures session keys. In most practical hybrid systems, the symmetric component is implemented using well-established algorithms such as AES without modifying their internal transformations. While this design ensures compatibility and reliability, the interaction between hybrid key encapsulation mechanisms and adaptive symmetric encryption behavior remains relatively underexplored in IoT-oriented cryptographic architectures [21].

Hybrid encryption has been widely adopted in secure communication systems to reduce computational overhead while maintaining strong cryptographic protection. In the KEM–DEM architecture, a fast symmetric cipher encrypts large volumes of data, while a public-key mechanism protects the symmetric session key. Several studies have explored hybrid encryption models combining AES with NTRU for secure communication in distributed and IoT environments. Empirical results indicate that AES–NTRU hybrid schemes can support real-time data transmission with acceptable latency and throughput. In these frameworks, AES is typically retained as the data encryption component due to its efficiency and widespread hardware support, while NTRU provides post-quantum resilience at the key encapsulation level [22]. However, most existing AES–NTRU hybrid frameworks employ the standard AES algorithm without modifying its internal structure [23]. As a result, research on hybrid encryption has largely focused on key management efficiency and system-level performance rather than exploring adaptive transformations within the symmetric encryption process.

Beyond hybrid encryption architectures, several studies have investigated modifications to the internal structure of symmetric encryption algorithms to improve diffusion behavior and resistance to statistical analysis. Approaches such as round-function transformations, key-dependent substitution mechanisms, and dynamic S-box constructions have been proposed to enhance the diffusion characteristics of AES-based systems, particularly in resource-constrained environments such as IoT and edge computing [24]. Dynamic S-box mechanisms introduce variability into the substitution stage by generating different substitution mappings across encryption rounds or sessions. These techniques aim to improve empirical diffusion characteristics, including avalanche behavior, while preserving the overall AES structure. Nevertheless, many dynamic AES variants rely on additional randomness sources or complex key-dependent transformations, which may introduce implementation complexity or synchronization requirements between communicating parties. To provide a clearer overview of existing research directions, table 1 summarizes representative studies according to their focus on key encapsulation mechanisms, hybrid encryption architectures, and adaptive symmetric encryption techniques.

**Table 1.** Classification of Related Work

| Study | Key Encapsulation (PQC / NTRU / Kyber) | Hybrid Encryption (KEM–DEM) | Adaptive Symmetric Encryption | Application Context |
|---|---|---|---|---|
| Ahmad et al. [14] | ✓ | ✓ | ✗ | Secure communication systems |
| Tiberti et al. [23] | ✓ | ✓ | ✗ | Automotive networks |
| Ehsan [11] | ✓ | ✓ | ✗ | IoT communication |

| | | | | |
|---|---|---|---|---|
| Ramakrishna [12] | ✓ | ✓ | ✗ | Embedded systems |
| Zahid et al. [25] | ✗ | ✗ | ✓ | AES modification |
| Devi and Kotha [24] | ✗ | ✗ | ✓ | AES transformation |
| Proposed Work | ✓ | ✓ | ✓ | IoT secure communication |

As shown in the table, most studies focus either on hybrid encryption architectures or on modifications to symmetric encryption algorithms independently. Hybrid encryption research typically emphasizes key protection and system-level efficiency, whereas adaptive symmetric encryption studies focus on modifying the internal behavior of algorithms such as AES.

## 2.2. Research Gap and Motivation

Existing literature can generally be categorized into three major research directions: (1) post-quantum key encapsulation mechanisms for IoT communication systems, (2) hybrid encryption architectures integrating symmetric and asymmetric cryptography, and (3) adaptive enhancement techniques applied to symmetric encryption algorithms [26], [27]. However, these research directions are typically investigated independently rather than within a unified cryptographic framework.

Specifically, many AES–NTRU hybrid encryption models emphasize secure key management and communication efficiency while preserving the standard AES internal structure. Conversely, studies on dynamic AES variants focus on modifying substitution behavior but often rely on conventional key protection mechanisms rather than hybrid architectures [28], [29]. This separation highlights a research gap between hybrid encryption frameworks and adaptive symmetric encryption mechanisms within integrated secure communication systems [30], [31]. Motivated by this gap, this study proposes a modified hybrid AES–NTRU encryption model that integrates a deterministic dynamic S-box into the AES encryption process. The proposed approach introduces controlled variability within the substitution stage while preserving compatibility with standard AES operations and maintaining the hybrid KEM–DEM architecture. By combining adaptive symmetric encryption behavior with post-quantum secure key encapsulation, the proposed model aims to provide an integrated cryptographic solution for secure IoT communication environments [32], [33], [34].

# 3. Methodology

This study proposes a hybrid cryptographic framework that integrates the Advanced Encryption Standard (AES) with the NTRU lattice-based cryptosystem within a Key Encapsulation Mechanism–Data Encapsulation Mechanism (KEM–DEM) architecture. The proposed design combines symmetric encryption for efficient data protection with post-quantum key encapsulation for secure session key exchange. In addition, a deterministic dynamic S-box mechanism is introduced into the AES substitution stage to introduce controlled variability during encryption while maintaining compatibility with the standard AES round structure. The overall architecture of the proposed system is illustrated in figure 1. The framework consists of three primary components: (1) AES-based payload encryption, (2) dynamic S-box generation derived from the AES key schedule, and (3) NTRU-based key encapsulation for secure transmission of the AES session key.

## 3.1. Hybrid AES–NTRU Encryption Architecture

In the proposed system, the encryption process follows the KEM–DEM paradigm. In figure 1, the plaintext data is encrypted using AES with a dynamically generated substitution layer. The AES session key used for this process is subsequently protected using the NTRU public-key cryptosystem. During transmission, the encrypted payload and the encapsulated session key are delivered together to the receiver. The receiver then performs NTRU decapsulation to recover the AES session key before executing the AES decryption process to restore the original plaintext. This architecture enables efficient data encryption while providing post-quantum-resistant protection for session keys.

ISSN 2723-6471


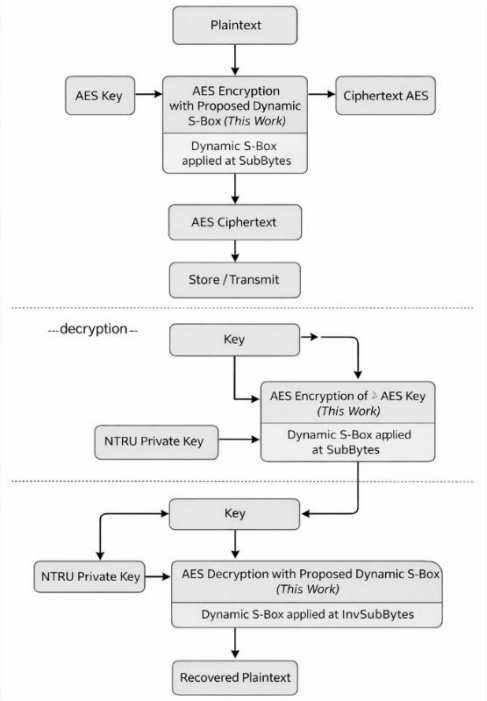


**Figure 1**. Hybrid AES–NTRU Encryption Framework

## 3.2. Deterministic Dynamic S-Box Construction

In standard AES, the SubBytes transformation uses a fixed substitution box S(·) that is reused across all encryption rounds. In the proposed method, a round-dependent dynamic substitution mechanism is introduced by applying a deterministic mask to the input byte before the standard AES S-box operation. For each encryption round *r*, the dynamic S-box $S_r(.)$ is defined as:

$$S_r(x) = S(x \oplus \delta_r) \tag{1}$$

Note:

$x \in \{0,1 \ldots.,255\}$ is the 8-bit input byte to be substituted,

$S(.)$ denotes the standard AES S-box,

$\oplus$ denotes the bitwise XOR operation, and

$\delta_r \in \{0,1 \ldots.,255\}$ is the round-dependent 8-bit mask for round *r*.

The mask $\delta_r$ is derived deterministically from the AES key schedule. The least significant byte of the first 32-bit word of the round key was selected as the mask source because it provides a deterministic, lightweight, and reproducible derivation rule. Accordingly, this choice should be understood as a practical design decision favoring simplicity and compatibility, rather than an assertion that it is the only or optimal derivation strategy. Let $RK_r$ denote the round key at round *r*, and let $RK_r[0]$denote the first 32-bit word of that round key. In this study, δr is defined as the least significant byte of $RK_r[0]$, i.e.,

$$\delta_r = K_r[0]_{LSB}) \tag{2}$$

$K_r$ denotes the AES round key at round *r*.

Since the standard AES S-box is bijective and XOR with a fixed mask is also bijective over 8-bit inputs, the resulting mapping $S(.)$ remains bijective. Therefore, the inverse dynamic S-box used during decryption is defined as:

$$S_r^{-1}(y) = S^{-1}(y) \oplus \delta_r \tag{3}$$

$y \in \{0,1, \ldots,255\}$ is the substituted output byte, and

$S^{-1}(.)$ denotes the standard AES inverse S-box.

This construction preserves invertibility and allows correct decryption, because for any input byte $x$,

$$S_r^{-1}(S_r(x)) = S^{-1}(S(x \oplus \delta r)) \oplus \delta_r = x \quad (4)$$

Thus, the proposed dynamic S-box does not replace the AES S-box with a newly designed lookup table; instead, it introduces a deterministic round-dependent masking of the S-box input while preserving the original AES substitution structure.

## 3.3. Encryption and Decryption Workflow

The operational workflow of the proposed encryption scheme is illustrated in figure 2. During the encryption phase, the plaintext payload is first encrypted using the AES algorithm with a dynamically generated S-box applied in the substitution stage. Subsequently, the AES session key used for the encryption process is encapsulated using the NTRU public-key cryptosystem following the Key Encapsulation Mechanism–Data Encapsulation Mechanism (KEM–DEM) framework. The resulting encrypted payload together with the encapsulated session key is then transmitted to the receiver. On the receiver side, the NTRU decapsulation process is performed to recover the AES session key, after which the AES decryption process is executed using the same dynamic substitution mapping derived deterministically from the round key to reconstruct the original plaintext.

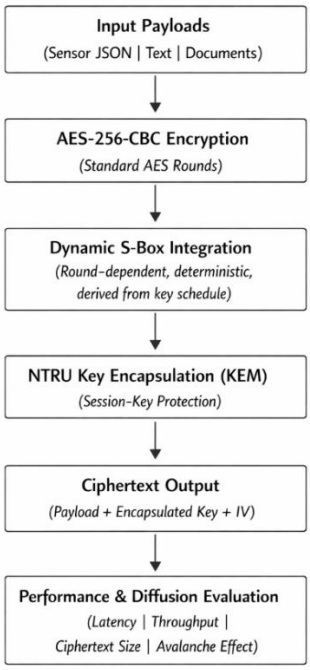


**Figure 2.** AES–NTRU hybrid modified model

## 3.4. Implementation Environment

The dynamic S-box is integrated into the AES SubBytes and InvSubBytes transformations. All other AES components, including ShiftRows, MixColumns, AddRoundKey, and the key schedule, remain unchanged [35] During encryption, the SubBytes operation applies the round-dependent dynamic S-box Sr(·) at each round. During decryption, the inverse dynamic S-box $S_r^{-1}$ is applied in the InvSubBytes operation. The proposed cryptographic framework was implemented using Python with the PyCryptodome cryptographic library to enable reproducible experimentation. The implementation focuses on validating the functional correctness of the hybrid architecture and evaluating the diffusion characteristics and computational performance of the proposed encryption scheme. The prototype environment does not employ hardware acceleration for AES operations; therefore, the measured performance results represent software-level evaluations suitable for prototype analysis.

## 4. Results and Discussion

The experimental evaluation aims to assess the functional correctness, computational performance, and empirical diffusion behavior of the proposed hybrid AES–NTRU encryption framework. All experiments were conducted using the Python-based prototype implementation described in Section 3. The objective of this evaluation is not to establish formal cryptographic security guarantees, but rather to examine the operational behavior and diffusion characteristics of the proposed architecture in a controlled experimental environment.

### 4.1. Experimental Setup

The prototype system was implemented using Python and the PyCryptodome cryptographic library. This environment was selected to facilitate reproducible experimentation and rapid validation of the hybrid encryption architecture. The experiments were conducted on a standard workstation platform without hardware acceleration for AES operations. Several representative IoT data payloads were used to evaluate the encryption process. These payloads simulate typical communication scenarios in IoT environments, where sensor readings or device messages are transmitted through gateway nodes. The evaluation focuses on three primary metrics to assess the effectiveness of the proposed encryption model. First, functional correctness is examined by verifying that all encrypted payloads can be successfully decrypted back to their original plaintext without any bit errors. Second, computational performance is evaluated by measuring the execution time required for both encryption and decryption processes, providing insights into the efficiency of the proposed scheme in practical implementations. Third, empirical diffusion behavior is analyzed using the avalanche effect metric, which assesses how small changes in the plaintext influence the resulting ciphertext, thereby indicating the diffusion characteristics of the encryption process.

### 4.2. Functional Correctness Verification

The correctness of the proposed hybrid encryption framework was validated by performing multiple encryption and decryption cycles on representative data payloads. In all experimental trials, the decrypted outputs matched the original plaintext inputs without any bit errors. This confirms that the integration of the dynamic S-box mechanism and the NTRU-based key encapsulation does not disrupt the functional integrity of the AES encryption–decryption process. The deterministic design of the dynamic S-box ensures that the same substitution mapping can be reproduced during decryption using the corresponding AES round keys. Consequently, both encryption and decryption processes remain fully synchronized without requiring additional parameter exchange between communicating parties.

### 4.3. Data Used in the Experiments

The evaluation was conducted using representative data payloads that reflect common communication patterns in IoT systems. Three categories of data were used in the experimental evaluation to represent typical payloads transmitted in IoT communication environments. The first category consists of sensor data, represented by small JavaScript Object Notation (JSON) formatted payloads that simulate periodic measurements generated by IoT devices. The second category includes text messages, which represent control commands and monitoring information exchanged between devices and gateway systems. The third category comprises document files, representing structured configuration data or report files that may be transmitted within IoT-based information systems. Payload sizes ranged from small to moderate, covering typical IoT communication workloads. The same datasets were used consistently across all experimental configurations to ensure fair comparison. Input Load Characteristics are shown in table 2.

**Table 2.** Input Payload Characteristics

| Payload Type | Avg Size (bytes) | Min–Max (bytes) | Messages |
|---|---|---|---|
| Sensor JSON | 256 | 128–512 | 1,000 |
| Text | 512 | 256–1,024 | 1,000 |
| Document | 4,096 | 2,048–8,192 | 500 |

### 4.4. Diffusion Behavior Analysis

To evaluate the diffusion characteristics of the proposed encryption scheme, the avalanche effect was measured by introducing a single-bit modification in the plaintext and observing the resulting change in the ciphertext. The avalanche effect indicates the proportion of output bits that change due to a small modification in the input.The experimental results show that the proposed hybrid encryption framework achieves an average avalanche effect of

47.01%. For comparison, standard AES implementations typically exhibit avalanche values close to 50% under ideal conditions.

Although the observed avalanche value is slightly lower than that of the standard AES implementation, the results still demonstrate consistent diffusion behavior within acceptable experimental bounds. This outcome suggests that the introduction of the deterministic dynamic S-box does not significantly degrade the diffusion properties of the AES encryption process while enabling adaptive substitution behavior within the hybrid framework. It is important to emphasize that the avalanche effect serves as an empirical indicator of diffusion behavior rather than a comprehensive measure of cryptographic strength. Formal cryptographic security evaluation would require additional analyses such as differential or linear cryptanalysis, which are beyond the scope of the present study.

## 4.5. Evaluated Encryption Configurations

To isolate and highlight the effect of the proposed dynamic S-box mechanism, three encryption configurations were evaluated in the experimental analysis. The first configuration is AES-only (static S-box), which applies standard AES-256 encryption using the original static substitution box and serves as the baseline configuration. The second configuration is an AES–NTRU hybrid with a static S-box, representing a conventional hybrid encryption scheme based on the KEM–DEM paradigm, where AES is used for payload encryption and NTRU is applied for session-key encapsulation without modifying the internal AES structure. The third configuration is the proposed AES–NTRU hybrid with a dynamic S-box, in which the AES encryption process is enhanced by integrating a deterministic round-dependent dynamic S-box into the SubBytes and InvSubBytes transformations while maintaining all other AES components unchanged. By comparing the results of configurations (2) and (3), the experiments explicitly evaluate the contribution of the dynamic S-box mechanism within the same hybrid encryption framework.

## 4.6. Evaluation Metrics

System performance was evaluated using several key metrics to assess the efficiency of the proposed encryption framework. These metrics include encryption time ($T_{Enc}$), which represents the time required to encrypt the payload using the AES algorithm; decryption time ($T_{dec}$), which measures the time required to recover the original plaintext from the ciphertext; and key encapsulation time ($T_{kem}$), which denotes the time required for the NTRU-based session-key encapsulation and decapsulation processes. In addition, the total cryptographic processing time ($T_{Total}$) was calculated as the overall time required to complete the entire encryption and decryption workflow within the hybrid cryptographic scheme.

$$T_{Total} = T_{Enc} + T_{dec} + T_{kem} \tag{5}$$

Throughput (τ) was calculated as the ratio of payload size to total cryptographic processing time:

$$\tau = \frac{Payload\ Size}{T_{Total}} \tag{6}$$

## 4.7. Implementation and Parameter Settings

All experiments were conducted on a gateway-level IoT platform equipped with an Intel x86-64 processor operating at 2.5–3.0 GHz under a 64-bit Linux environment. The implementation was developed using Python 3.10 with the PyCryptodome library (version 3.x) for AES operations and publicly available Python bindings of the NTRU reference implementation. No hardware acceleration, parallel execution, or library-specific optimizations were enabled, ensuring consistent evaluation across all configurations.

All evaluated schemes used identical AES parameters to maintain fair comparison: AES-256 with Cipher Block Chaining (CBC) mode, PKCS#7 padding, and a randomly generated 128-bit initialization vector for each encryption session. Each message employed a unique ephemeral AES session key. The proposed Dynamic S-Box modification was applied only to the SubBytes and InvSubBytes transformations, while all other AES components (ShiftRows, MixColumns, AddRoundKey, and key schedule) remained identical to the standard AES specification. For each AES round $r$, the round-dependent mask $\delta_r$ was derived deterministically from the expanded AES key schedule as the least significant byte of the first 32-bit word of the corresponding round key. This deterministic rule allows both encryption and decryption processes to regenerate identical Dynamic S-Boxes without exchanging additional parameters while preserving bijectivity. NTRU was used exclusively as a Key Encapsulation Mechanism (KEM) to protect the AES

session key, while payload data remained encrypted using AES. A fixed NTRU parameter profile providing post-quantum security comparable to approximately 128-bit classical security was used consistently across all experiments to ensure a fair evaluation. Three encryption configurations were examined in the study. The first configuration was AES-only with a static S-box, representing the standard AES encryption used as the baseline. The second configuration was AES–NTRU hybrid encryption with a static S-box, which follows the conventional KEM–DEM hybrid architecture where AES encrypts the payload and NTRU encapsulates the session key without modifying the AES internal structure. The third configuration was the proposed AES–NTRU hybrid scheme employing a deterministic round-dependent dynamic S-box, in which the substitution stage of AES is enhanced while maintaining compatibility with the standard AES round structure.

## 4.8. Encryption and Decryption Performance

Encryption and decryption performance tests were conducted using representative IoT communication data, including sensor data in JavaScript Object Notation (JSON) format, short text messages, and structured document files. These datasets were selected to reflect common IoT workloads and were applied consistently across all experimental configurations. The test data were evaluated under three different encryption conditions to analyze the impact of the proposed mechanism. The first condition employed an AES-only configuration, representing the standard AES encryption scheme used as the baseline. The second condition applied an AES–NTRU hybrid encryption with static S-boxes, following the conventional KEM–DEM architecture where AES encrypts the payload while NTRU encapsulates the session key without modifying the internal AES substitution structure. The third condition implemented the AES–NTRU hybrid encryption with dynamic S-boxes, in which the proposed deterministic round-dependent dynamic substitution mechanism is integrated into the AES encryption process to examine its effect on the overall cryptographic behavior. The results for these three tests are shown in table 3.

**Table 3.** Comparison of the Advantages of Three Encryption Models

| Encryption Model | Avalanche Effect | Encryption Time | Decryption Time | Key Validation (PGP) |
|---|---|---|---|---|
| AES-only configuration | 52.46% | 49.55 ms | 1.35 ms | Not Applicable |
| AES–NTRU with Static S-boxes | 51.41% | 69.47ms | 9.19 ms | Successful |
| AES–NTRU with Dynamic S-boxes | 47.01% | 118.99 ms | 5.91 ms | Successful |

The comparison in table 3 presents the diffusion behavior and computational performance of the three evaluated encryption configurations. The AES-only configuration, used as the baseline, achieves the highest avalanche effect (52.46%) with very fast decryption time, reflecting the strong diffusion properties and efficiency of the standard AES substitution mechanism. The AES–NTRU configuration with a static S-box introduces hybrid key encapsulation while preserving the standard AES structure. This configuration maintains a comparable avalanche effect (51.41%) while providing secure session-key protection through NTRU encapsulation and PGP-based validation. The slightly increased execution time is mainly caused by the additional operations required for key encapsulation and verification.

The proposed AES–NTRU model with a deterministic dynamic S-box introduces round-dependent variability within the AES substitution stage. The experimental results show an avalanche effect of 47.01%, which is slightly lower than the baseline AES implementation. This indicates that the dynamic masking mechanism does not aim to enhance the intrinsic diffusion properties of AES, but rather introduces controlled structural variability while maintaining compatibility with the AES architecture and the hybrid KEM–DEM framework. Despite the lower avalanche value, the proposed scheme maintains correct encryption–decryption behavior and acceptable computational overhead, making it suitable for gateway-level IoT communication environments.

## 4.9. Throughput Analysis

The throughput experiment evaluates the computational characteristics of the evaluated encryption configurations under identical input conditions. The results summarized in table 4 show that the baseline AES ciphertext configuration achieves the fastest encryption and decryption performance due to the absence of hybrid key encapsulation operations. The AES Cipherkey configuration exhibits significantly higher encryption time because the encryption process is applied directly to key-level data structures, which increases computational overhead. The AES Ciphertext + AES Cipherkey configuration represents a balanced hybrid scheme where payload encryption and key protection are combined, resulting in moderate computational cost while maintaining strong diffusion properties.

**Table 4.** Performance Summary of Each Encryption Model

| Encryption Model | Plaintext Size (bytes) | Ciphertext Size (bytes) | Avalanche (%) | Encryption Time (ms) | Decryption Time (ms) |
|---|---|---|---|---|---|
| AES Ciphertext | 71 | 96 | 52.46 | 49.55 | 1.35 |
| AES Cipherkey | 71 | 64 | 0.00 | 130.96 | 9.25 |
| AES Ciphertext + AES Cipherkey | 71 | 160 | 51.41 | 69.47 | 9.19 |
| Hybrid Super Encryption | 71 | 160 | 47.01 | 118.99 | 5.91 |

## 5. Conclusion

This study was motivated by a research gap identified in existing hybrid encryption systems based on the KEM–DEM paradigm, where the primary focus is placed on key protection and system-level efficiency, while the symmetric encryption component is commonly employed in its standard, static form. As discussed in the related work, limited attention has been given to adaptive mechanisms within symmetric encryption, particularly regarding substitution and diffusion behavior in complete end-to-end encryption systems. To address this gap, this research proposed a hybrid encryption model that integrates AES for payload encryption and NTRU with Pretty Good Privacy (PGP) for secure key encapsulation, enhanced by a deterministic round-dependent dynamic S-box. Unlike conventional hybrid schemes, the proposed approach introduces controlled adaptation within the AES SubBytes and InvSubBytes transformations, while preserving the original AES round structure, key schedule, and compatibility. The deterministic design ensures synchronization between encryption and decryption without requiring additional metadata exchange.

The experimental results demonstrate that the proposed hybrid AES–NTRU architecture successfully integrates deterministic dynamic substitution behavior within the AES encryption process while maintaining compatibility with the standard AES round structure. Although the avalanche effect of the proposed configuration (47.01%) is slightly lower than that of the baseline AES implementation (52.46%), the mechanism introduces controlled substitution variability without significantly degrading diffusion characteristics. The results indicate that the proposed approach can support hybrid encryption architectures with acceptable computational overhead while enabling structural diversity in symmetric encryption processes.

## 6. Declarations

### 6.1. Author Contributions

Conceptualization: P.S., P.S., M.A.B., and J.T.T.; Methodology: P.S.; Software: P.S.; Validation: P.S., P.S., M.A.B., and J.T.T.; Formal Analysis: P.S., P.S., M.A.B., and J.T.T.; Investigation: P.S.; Resources: P.S.; Data Curation: P.S.; Writing Original Draft Preparation: P.S., P.S., M.A.B., and J.T.T.; Writing Review and Editing: P.S., P.S., M.A.B., and J.T.T.; Visualization: P.S.; All authors have read and agreed to the published version of the manuscript.

### 6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

### 6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

### 6.4. Institutional Review Board Statement

Not applicable.

### 6.5. Informed Consent Statement

Not applicable.

### 6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] Y. Bin Zikria, R. Ali, M. K. Afzal, and S. W. Kim, "Next-Generation Internet of Things (IoT): Opportunities, Challenges, and Solutions," *Sensors*, vol. 21, no. 4, p. 1174, pp. 1-7, 2021, doi: 10.3390/s21041174.

[2] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial internet of things: Recent advances, enabling technologies and open challenges," *Computers & Electrical Engineering*, vol. 81, no. January, art. no. 106522, pp. 1-24, 2020, doi: 10.1016/j.compeleceng.2019.106522.

[3] V. Verma, P. Kumar, R. K. Verma, and S. Priya, "A Novel Approach for Security in Cloud Data Storage Using AES-DES-RSA Hybrid Cryptography," in *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, Raigarh, India, IEEE, vol. 2021, no. December, pp. 1–6, 2021, doi: 10.1109/ETI4.051663.2021.9619274.

[4] E. Jintcharadze and M. Iavich, "Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems," in *2020 IEEE East-West Design & Test Symposium (EWDTS)*, Varna, Bulgaria, IEEE, vol. 2020, no. October, pp. 1–5, 2020, doi: 10.1109/EWDTS50664.2020.9224901.

[5] G. Dhamodharan, "An Enhanced and Dynamic Key AES Algorithm for Internet of Things Data Security," *Journal of Advanced Zoology*, vol. 44, no. S-6, pp. 1323–1332, 2023, doi: 10.17762/jaz.v44iS6.2444.

[6] A. Hamza and B. Kumar, "A Review Paper on DES, AES, RSA Encryption Standards," in *Proceedings of the SMART–2020*, Moradabad, India, IEEE, vol. 2021, no. February, pp. 333–338, 2020, doi: 10.1109/SMART50582.2020.9336800.

[7] A. Umrani, A. K. Vangujar, and P. Palmieri, "mKEM-DEM Based Anonymous Certificateless Hybrid Signcryption for Broadcast VANET Communication," in *Information Systems Security and Privacy*, Springer Nature Switzerland, vol. 2025, no. July, pp. 246–273, 2024, doi: 10.1007/978-3-031-89518-0_12.

[8] R. Chen, X. H. B, and M. Yung, "Subvert KEM to Break DEM: Practical Algorithm-Substitution Attacks on Public-Key Encryption," in *Advances in Cryptology – ASIACRYPT 2020*, Springer International Publishing, vol. 2020, no. December, pp. 98–128, 2020, doi: 10.1007/978-3-030-64834-3_4.

[9] P. He and M. Xu, "An NTRU-Based Key Encapsulation Scheme for Underwater Acoustic Communication," *Electronics*, vol. 14, no. 3, art. no. 405, pp. 1-17, 2025, doi: 10.3390/electronics14030405.

[10] M. A. Al-Shabi, "A Survey on Symmetric and Asymmetric Cryptography Algorithms in Information Security," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 3, art. no. 8779, pp. 1-14, 2019, doi: 10.29322/ijsrp.9.03.2019.p8779.

[11] M. A. Ehsan, W. Alayed, A. U. Rehman, W. ul Hassan, and A. Zeeshan, "Post-Quantum KEMs for IoT: A Study of Kyber and NTRU," *Symmetry*, vol. 17, no. 6, pp. 1–13, 2025, doi: 10.3390/sym17060881.

[12] D. Ramakrishna and M. A. Shaik, "PSESV: A hybrid post-quantum encryption Framework with real-time thermal and EM side-channel attack detection," *Ain Shams Engineering Journal*, vol. 16, no. 12, art. no. 103776, pp. 1-10, 2025, doi: 10.1016/j.asej.2025.103776.

[13] S. Sharifian and R. Safavi-Naini, "Information-theoretic Key Encapsulation and its Application to Secure Communication," in *2021 IEEE International Symposium on Information Theory (ISIT)*, IEEE, vol. 2021, no. July, pp. 2393–2398, Jul. 2021, doi: 10.1109/ISIT45174.2021.9517958.

[14] K. Ahmad, A. Kamal, K. A. Bin Ahmad, M. Khari, and R. G. Crespo, "Fast hybrid-MixNet for security and privacy using NTRU algorithm," *Journal of Information Security and Applications*, vol. 60, no. August, art. no. 102872, pp. 1-14, 2021, doi: 10.1016/j.jisa.2021.102872.

[15] A. Dutta, R. Bose, S. Roy, and S. Sutradhar, "Hybrid Encryption Technique to Enhance Security of Health Data in Cloud Environment," *Archives of Pharmacy Practice*, vol. 14, no. 3, pp. 41–47, 2023, doi: 10.51847/raeh8fHBt6.

[16] E. Gelenbe, M. Nakip, D. Marek, and T. Czachorski, "Diffusion Analysis Improves Scalability of IoT Networks to Mitigate the Massive Access Problem," in *Proceedings - IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS)*, IEEE, vol. 2021, no. November, pp. 1–8, 2021, doi: 10.1109/MASCOTS53633.2021.9614289.

[17] A. Čolaković, "IoT systems modeling and performance evaluation," *Computer Science Review*, vol. 50, no. November, p. 100598, pp. 1-26, Nov. 2023, doi: 10.1016/j.cosrev.2023.100598.

[18] D. Liestyowati, "Public Key Cryptography," *Journal of Physics: Conference Series*, vol. 1477, no. 5, p. 052062, pp. 1-8, Mar. 2020, doi: 10.1088/1742-6596/1477/5/052062.

[19] A. M. Qadir and N. Varol, "A review paper on cryptography," in *7th International Symposium on Digital Forensics and Security (ISDFS 2019)*, IEEE, vol. 2019, no. July, pp. 1–6, 2019, doi: 10.1109/ISDFS.2019.8757514.

[20] R. Akter, M. A. R. Khan, F. Rahman, S. J. Soheli, and N. J. Suha, "RSA and AES Based Hybrid Encryption Technique for Enhancing Data Security in Cloud Computing," *International Journal of Computational and Applied Mathematics & Computer Science*, vol. 3, no. October, pp. 60–71, 2023, doi: 10.37394/232028.2023.3.8.

[21] M. J. Alanazi, R. A. Alhoweiti, G. A. Alhwaity, and A. R. Alharbi, "An Adaptive Hybrid Cryptographic Framework for Resource-Constrained IoT Devices," *Electronics*, vol. 14, no. 23, p. 4666, pp. 1-23, 2025, doi: 10.3390/electronics14234666.

[22] H. Wei et al., "Efficient Implementation of NTRU-Based Key Encapsulation Mechanism on Embedded Platform," in *Data Security and Privacy Protection*, vol. 2025, no. October, pp. 412–431, 2025, doi: 10.1007/978-981-95-3182-0_26.

[23] W. Tiberti, R. Civino, and M. Pugliese, "A Hybrid-Cryptography Engine for Securing Intra-Vehicle Communications," *Applied Sciences*, vol. 13, no. 24, pp. 1–20, 2023, doi: 10.3390/app132413024.

[24] S. Devi and H. D. Kotha, "AES encryption and decryption standards," in *International Conference on Computer Vision and Machine Learning*, vol. 2018, no. December, pp. 1–11, 2018, doi: 10.1088/1742-6596/1228/1/012006.

[25] A. H. Zahid, H. Rashid, M. M. U. Shaban, S. Ahmad, E. Ahmed, M. T. Amjad, M. A. T. Baig, M. J. Arshad, M. N. Tariq, M. W. Tariq, M. A. Zafar, and A. W. Basit, "Dynamic S-Box Design Using a Novel Square Polynomial Transformation and Permutation," *IEEE Access*, vol. 9, no. June, pp. 82390–82401, 2021, doi: 10.1109/ACCESS.2021.3086717.

[26] D. Shivaramakrishna and M. Nagaratna, "A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control," *Alexandria Engineering Journal*, vol. 84, no. December, pp. 275–284, 2023, doi: 10.1016/j.aej.2023.10.054.

[27] S. Rehman, N. Talat Bajwa, M. A. Shah, A. O. Aseeri, and A. Anjum, "Hybrid AES-ECC Model for the Security of Data over Cloud Storage," *Electronics*, vol. 10, no. 21, art. no. 2673, pp. 1-20, 2021, doi: 10.3390/electronics10212673.

[28] H. Chien, "Dynamic Public Key Certificates with Forward Secrecy," *Electronics*, vol. 10, no. 16, art. no. 2009, pp. 1-14, 2021, doi: 10.3390/electronics10162009.

[29] Y. Watanabe, K. Ohara, M. Iwamoto, and K. Ohta, "Efficient Dynamic Searchable Encryption with Forward Privacy under the Decent Leakage," in *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, vol. 2022, no. April, pp. 312–323, 2022, doi: 10.1145/3508398.3511521.

[30] H. T. Assafli, I. A. Hashim, and A. A. Naser, "Advanced Encryption Standard (AES) acceleration and analysis using graphical processing unit (GPU)," *Applied Nanoscience*, vol. 13, no. 2, pp. 1245–1250, Feb. 2023, doi: 10.1007/s13204-021-01985-3.

[31] P. Elumalaivasan, T. Munirathinam, V. Kayalvizhi, G. Sekar, T. M. Sivanesan, and S. G, "Comparative Analysis of AES and AES-RSA Hybrid Techniques for Securing Visual Data Integrity," in *11th International Conference on Communication and Signal Processing (ICCSP)*, IEEE, vol. 2025, no. July, pp. 870-875, 2025, doi: 10.1109/ICCSP64183.2025.11089233.

[32] S. B. Basapur, B. S. Shylaja, and Venkatesh, "A Hybrid Cryptographic Model Using AES and RSA for Sensitive Data Privacy Preserving," *Webology, Current Trends in Management and Information Technology*, vol. 18, no. Special Issue, pp. 129–148, 2021, doi: 10.14704/WEB/V18SI05/WEB18219.

[33] S. Fatima, T. Rehman, M. Fatima, S. Khan, and M. A. Ali, "Comparative Analysis of AES and RSA Algorithms for Data Security in Cloud Computing," in *Engineering Proceedings*, vol. 20, no. 1, pp. 1-6, 2022, doi: 10.3390/engproc2022020014.

[34] S.-W. Lee and K.-B. Sim, "Design and Hardware Implementation of a Simplified DAG-Based Blockchain and New AES-CBC Algorithm for IoT Security," *Electronics*, vol. 10, no. 9, p. 1127, pp. 1-20, May 2021, doi: 10.3390/electronics10091127.

[35] Azanuddin, A. H. Nasyuha, I. Ruslianto, Moch. I. P. Angin, M. H. Aly, and M. A. Agoi, "RSA-AES Cryptosystem with Auto-Key Rotation for Cloud Storage," *Zero: Jurnal Sains, Matematika, dan Terapan*, vol. 9, no. 3, pp. 1031–1042, 2025, doi: 10.30829/zero.v9i3.26827.