

Original Article

PQC-Guard: A Unified Cryptographic Framework Integrating Hybrid Encryption and Key Exchange for Comprehensive Post-Quantum Data Security

Dasari Veera Reddy¹, Padmaja Madugula²

^{1,2}Department of Computer Science and Engineering, GSCSE, GITAM Deemed to be University, Visakhapatnam, India.

¹Corresponding Author: vrdasari@gitam.in

Received: 27 July 2026

Revised: 02 October 2026

Accepted: 14 October 2026

Published: 28 October 2026

Abstract - The advent of quantum computing threatens conventional public-key cryptographic systems, and there is an urgent need for comprehensive frameworks to aid the shift towards post-quantum cryptography. Current approaches that establish a key and encrypt data separately suffer from disjointed protection, limited protection capabilities, integration complexity, and inadequate consideration of the performance-security relationship. In this study, we propose a hybrid cryptographic framework, PQC-Guard, for post-quantum-oriented information security that combines hybrid key establishment and multilayer encryption. The proposed GuardFusionAlgo is a hybrid key establishment (as DSKE), hybrid session-key derivation, encryption-specific sub-key generation, multilayer protection (as MDES), QBM transformations, secure transmission and reverse-order data recovery. Experimental evaluation shows that PQC-Guard has a ciphertext entropy of 7.997 bits/byte, an avalanche effect of 50.21%, a key sensitivity of 50.18%, a plaintext-ciphertext correlation of 0.0012, a 100% tampering detection rate, a 99.6% replay detection rate, and a 98.9% partial key-compromise resistance rate. The framework yields an end-to-end throughput of 15.72 MB/s for a 500 MB workload, negligible ciphertext expansion, manageable computational resource consumption, and 100% successful recovery in all experiments conducted concurrently. Ablation and comparative analyses of the individual components, along with statistical analyses, further highlight each component's contribution and confirm a strong overall performance-security trade-off. PQC-Guard offers a viable solution for post-quantum-ready cloud storage, enterprise communication, secure file transfer, and distributed data protection environments.

Keywords - Post-Quantum Cryptography, Hybrid key exchange, Multilayer encryption, Cryptographic security, Quantum-Resistant data protection.

1. Introduction

Rapid advances in quantum computing have posed substantial challenges to traditional cryptographic systems that safeguard sensitive data across cloud computing, enterprise communications, financial applications, healthcare networks, and distributed infrastructures. A sufficiently large quantum computer, capable of efficiently running quantum algorithms, could break public-key cryptographic systems like RSA and elliptic-curve cryptography [1, 2]. As a result, Post-Quantum Cryptography (PQC) has become an important research area for designing security mechanisms against both classical and quantum threats [3]. Recent studies have explored lattice-based cryptography, code-based schemes, hash-based signatures, isogeny-based mechanisms, and hybrid classical/post-quantum approaches [4, 5]. Hybrid cryptographic systems are among the most crucial systems during the shift from classical to quantum cryptography, as they use both classical- and quantum-resistant methods [6]. Several previous works have suggested hybrid key exchange

protocols, quantum-safe key establishment protocols and multi-layer encryption protocols [7-10]. Current solutions, however, tend to be tailored to each security function and lack built-in end-to-end mechanisms. Multilayer encryption techniques mainly focus on providing a high level of confidentiality while relying on other key-exchange schemes to create secrets, whereas key-exchange schemes mainly establish shared secrets. Furthermore, significant overhead in computation and communication, coordination problems within cryptographic components, as well as scalability and/or component-wise contribution to security, may arise in existing hybrid settings.

An integrated framework that is able to coordinate hybrid key establishment, generate secure session keys, derive keys for specific data, protect data by a multilayer structure, transform the data so that it can be reverted, securely transport the data, and reliably recover the data within one operational process is needed. However, three research gaps remain. First,



in the current landscape of hybrids, key establishment and end-to-end data protection are still not integrated seamlessly as a single operation – the coordination and operation of the cryptographic elements still represent just a distinct design issue. Second, few studies have focused on building a common session-key management mechanism that can generate and separate encryption-specific keys from various key-establishment processes while maintaining clear separation of cryptographic activities. Third, little systematic assessment so far exists on how much different cryptographic parts contribute to the performance, scalability, security-related properties and computational overhead. In particular, the combined effect of hybrid key derivation, multi-layer encryption, and reversible ciphertext transformation is not sufficiently analysed when components are ablated and repeated runs are evaluated using statistical analysis. The challenges arising in these gaps provide an impetus to the design and experimental analysis of a single architecture that tackles both key establishment and data protection as well as ciphertext transformation and recovery, along with the introduction of the performance-security trade-offs from each of these components.

The proposed PQC-Guard framework combines all four functions into a coordinated end-to-end system, rather than analysing them individually as hybrid key establishment, post-quantum secure communication, cloud-oriented protection, and multilayer encryption, as in existing studies. The novelty of PQC-Guard is its architectural integration of dual-source key establishment, hybrid session-key generation, sub-key generation for encryption, multi-layer data protection, reversible ciphertext transformation, secure data transmission, and reverse-order data recovery. Moreover, the study assesses each component's contribution by removing it from the framework in an ablation study, and it also evaluates the full framework against classical, hybrid, and post-quantum-oriented setups through repeated experiments and statistical analysis. Section 2 details the differences between the present approaches and PQC-Guard.

To address these gaps, this work proposes and empirically tests an integrated cryptographic architecture (PQC-Guard) focused on post-quantum-oriented data security. It comprises two key mechanisms: hybrid key establishment, implemented by a scheme called Dual-Source Key Establishment (DSKE), and multilayer data encryption, implemented according to the Multilayer Data Encryption Scheme (MDES). The proposed GuardFusionAlgo handles hybrid key establishment, session-key generation, sub-key generation for specific encryption algorithms, multilayer encryption, reversible transformation using QBM, secure transmission, and reverse-order data recovery. The framework addresses identified gaps in integration, key management, and evaluation through a common operational workflow and by enabling evaluation of security- and performance-related properties on a component basis. The main contributions of the framework are: (i) unified

coordination of hybrid key establishment and data protection; (ii) dual-source session-key construction; (iii) encryption-specific sub-key generation and separation; (iv) integrated multilayer data protection; (v) reversible QBM-based ciphertext transformation; and (vi) systematic evaluation of the performance-security trade-offs associated with the individual components.

This study analyses its contributions through a thorough experimental protocol covering key-establishment performance, encryption and decryption efficiency, scalability, communication and storage overhead, computational resource utilisation, and security-related statistical properties. A controlled ablation study assesses the contribution of individual components: hybrid key derivation, RSA key protection, and QBM transformation. Moreover, we run repeated experiments for PQC-Guard and report confidence intervals, statistical significance tests, and effect-size calculations, comparing it with classical, hybrid, and PQ-oriented configurations. This assessment is obtained quantitatively in terms of the computation and security aspects of the proposed integrated architecture.

Hence, more than the cryptographic primitives alone could be the problem to be solved, because of the necessity in a hybrid system to design an integrated architecture for hybrid key establishment, secure session-key generation, multiple levels of data protection, reversible mapping of the encrypted data and data recovery and thus quantify how much each part contributes to the Design Problem in terms of security and performance. This paper is organised as follows. Section 2 presents a review of related work, covering classical key exchange, post-quantum key exchange, hybrid key exchange, and integrated data-security mechanisms, and identifies research gaps. Section 3 details the proposed PQC-Guard methodology, including GuardFusionAlgo, the mathematical formulation, the architectural design, the operational workflow, the security mechanisms, and the algorithmic representation. Section 4 presents experimental results on performance, scalability, overhead, security evaluation, ablation studies, comparative analysis, and statistical validation. Section 5 discusses the main findings, research implications, and study limitations. Finally, Section 6 summarises the work and outlines future directions for the standardised integration of PQC, formal security analysis, side-channel protection, and deployment.

2. Related Work

The shift to post-quantum data security has spurred extensive research into hybrid cryptographic systems that integrate classical, post-quantum, and quantum-assisted components to maintain compatibility with existing systems while mitigating quantum threats. Early studies on hybrid key-combination formulations have highlighted the possibility of redundancy against heterogeneous attack models: if several independent assumptions are violated, the confidentiality of

the key combinations is not guaranteed [1, 3, 7, 8]. Other hybrid quantum-classical protocol designs have been explored for secure network migration, focusing on compatibility between the new and current networks, gradual deployment, and enhanced resistance to both classical and quantum attacks [4, 9]. These studies lay the groundwork for unified approaches that rely not on a single cryptographic primitive but on complementary methods for long-term protection.

Much of the literature focuses on post-quantum authentication and key exchange for communication protocols. This paper explores hybrid PQC-based authentication and key agreement for 6G, IoT, and ultra-low-latency settings, which are susceptible to quantum attacks and face replay, impersonation, and tracking threats under traditional methods like RSA and ECC. TLS-oriented studies also explain how PQC and QKD can be concatenated and combined with XOR, and they require rethinking the TLS handshake; however, they also highlight latency and key-material size issues for deployment [10, 23]. The following protocol-level adaptations have been proposed as a possible solution to hybrid or full post-quantum secure channels: WireGuard, eSIM secure channels, end-to-end messaging, email communication, mobile device communication, and other protocols; hybrid or full post-quantum secure channels is possible, but need careful optimization for bandwidth, runtimes, device constraints, and backward compatibility [20, 22, 26, 30].

Much focus has shifted to cloud, edge, and multi-user systems because they store long-term, high-value data that can be subject to “harvest-now, decrypt-later” attacks. This is because several works [5, 13, 14, 17, 25, 29] have been proposed to ensure the confidentiality, access control, and auditability of cloud data using PQC, blockchain, decentralised cloud nodes, multilayer encryption, and adaptive key management. The studies focused on edge devices emphasise lightweight PQC schemes, efficient key exchange, and compact implementations, with lattice-based schemes often favoured for their synergies in security, speed, and practical implementation [24, 27]. Multi-cloud and federated architectures take this trend further by incorporating quantum-resistant encryption, zero-trust policies, adaptive session control, AI-driven anomaly detection, and automated key lifecycle management [31, 32, 35, 36, 38]. These efforts highlight the need for a unified security model and a comprehensive cryptographic architecture to secure data storage, transmission, and cross-domain communication.

Another field of research is hardware and entropy-aware implementations. Researchers have implemented hybrid PQC on FPGAs and embedded platforms, showing it can be practically realised with reasonable area, latency, and memory overhead. High-quality entropy sources, such as TRNGs, can improve key-generation reliability [12, 19]. Other hybrid chaos-based quantum-enhanced AES schemes focus on high-

entropy key generation, frequent key changes, and symmetric encryption resistance against Grover-type search attacks [6, 28]. Meanwhile, quantum-resilient security controls show how PQC can be used to enable secure boot, secure update, secure access, and secure TLS with standardised candidates such as Kyber, Dilithium, Falcon, XMSS, and SPHINCS+ [16, 21].

Recent surveys and road maps highlight that post-quantum migration is not just a cryptographic replacement problem, but also an architectural, operational and governance challenge [18, 33, 34, 37]. The QKD review demonstrates its scalability, long-distance capabilities, and cost limitations, as well as certification and interoperability issues [15]. In summary, existing studies support the promise of hybrid encryption, PQC-based key encapsulation, PQC-enabled adaptive security, and AI-assisted keying by QKD. However, many examples address only parts of the system, such as TLS, messaging, cloud storage, hardware acceleration, or migration in a particular sector. Thus, a common framework like PQC-Guard is needed to integrate hybrid encryption, post-quantum key exchange, adaptive key management, and deployment-aware security controls into a single holistic architecture to ensure data security in the post-quantum era.

2.1. Research Gap, Novelty, and Comparison with Existing Research

Pre-existing works provide several significant directions for hybrid as well as post-quantum security. Studies involving hybrid key-combination have been used to investigate different combinations of classical and quantum-resistant key-combination, as a redundancy measure against various cryptographic threats [1, 3, 7, 8]. There are other hybrid quantum-classical secure migration solutions and compatibility between legacy and quantum infrastructure proposal studies [4, 9]. Post-quantum authentication and key agreement have been studied further to secure communications in 6G, IOT, and low-latency networks, and studies in post-quantum TLS have investigated the combining of PQC and QKD in handshakes in the context of communications [10, 11, 23]. These studies present evidence of the necessity of hybrid key establishment and interoperability and migration-aware security mechanisms.

Post quantum protection for cloud, edge, multi-user and distributed environments are also investigated in existing research works. With respect to cloud-oriented studies, they have taken into account factors such as confidentiality [5], access control [14] and auditability [17] and protection through blockchains [25] and multilayer encryption [29], and adaptive key management. Research on Edge has focused on lightweight Post-Quantum options, an efficient approach to key exchange, and compact implementation [2, 24, 27]. Hybrid PQC systems working with hardware have shown how to implement PQC mechanisms on FPGA and embedded devices, such as SIKE and ECDH systems, and entropy-aware

implementations have studied how to implement an entropy generation mechanism from hardware. They show that post-quantum security has been studied in various deployment contexts, but most post-quantum approaches focus on a single protocol, deployment context, cryptographic primitive, and/or security function without coordinating the full key-establishment-to-data-recovery process in a single experimental construction.

From these findings, this study addresses a threefold research gap. First, existing studies mainly focus on hybrid key establishment, secure communication protocols, cloud protection, or multilayered encryption as standalone methods; only a few studies coordinate and co-integrate these methods into a single end-to-end data-security workflow. Second, current methods give little attention to the generation of a common key for all the key establishment input sources, and to the further creation of the different keys for separate encryption and transformation steps. Third, while some previous works cite the benefits of security and performance of single approaches, systematic component-wise analysis of security and performance gains provided by hybrid key derivation, multilayer encryption and reversible ciphertext transformation is still limited. Thus, an integrated design is needed where key establishment, generation of session keys, encryption of data, transformation of ciphertext, secure transmission and recovery, as well as component-wise evaluation are all considered as related operations.

The novelty of PQC-Guard lies, hence, not in new cryptographic primitives per se but is of an architectural and evaluative nature. The framework provides a mechanism called GuardFusionAlgo for the coordination of DualSecure Key Exchange (DSKE) and Multilayered Dynamic Encryption Scheme (MDES). DSKE is a merging of ECDH and SIDH-derived shared secrets, followed by a merging via HKDF to create a single session key. The derived key for the session is used to generate more sub-keys to be used in the downstream MDES operation. In the proposed processing

sequence, AES, RSA, ChaCha20, and the reversible QBM transformation are then coordinated by MDES. By doing so, it produces one operational path from key establishment to encryption, on to transformation of the ciphertext, transmission and, last, recovery followed in reverse order.

The second novelty is evaluation by component in a systematic manner. The study does not assess the whole framework, but rather removes or isolates some of the individual components: hybrid key exchange, ECDH-only key derivation, SIDH-only key derivation, RSA layer, and QBM transformation. This allows us to observe the impact of each part on the entropy, avalanche effect, plaintext-ciphertext correlation, tamper detection, replay detection and resistance to partial key compromise.

A third novelty is evaluating classical, hybrid, and post-quantum-oriented configurations within a common experimental protocol. All evaluated configurations are either PQC-Guard or a mix of PQC and AES, with the former serving as the Primary Additive and the latter as the Secondary Additive. We measured performance and security characteristics under the same experimental conditions, repeating experiments for all approaches using a 100 MB workload.

Thus, PQC-Guard is not claimed to be a contribution resulting from replacing well-known cryptographic primitives with newly invented ones. The contribution, rather, is the coordination of heterogeneous cryptographic mechanisms in a single architecture, the creation of a common session-key/sub-key management workflow, and the systematic quantification of the performance-security impact of the integrated components.

Table 1 summarises the differences between representative existing research directions and the proposed PQC-Guard framework.

Table 1. Comparison of existing research with PQC-Guard

Approach	Main Focus	Limitation	PQC-Guard Contribution
Hybrid PQC [1, 7]	Hybrid key establishment	Mainly focuses on key exchange	Integrates key establishment with data protection
PQC-TLS [10, 23]	Secure communication and key exchange	Focuses mainly on transport/handshake	Provides end-to-end encryption and recovery
Cloud PQC [5, 13]	Secure cloud data	Focuses on cloud-specific protection	Combines hybrid keys with multilayer encryption
Multilayer Encryption [25]	Multiple encryption layers	Does not integrate hybrid key establishment	Integrates DSKE with MDES and QBM
PQC-Guard (Proposed)	Unified key establishment and data protection	-	GuardFusionAlgo + DSKE + MDES + QBM + ablation and comparative evaluation

3. Proposed Framework

This section presents the proposed PQC-Guard framework for post-quantum-oriented data security. It introduces the system architecture, the mechanism for GuardFusionAlgo integration, multiple-layer encryption based on MDES, the QBM transformation, and the secure data recovery process. The section also mathematically codifies the framework, conducts a security analysis, outlines an operational workflow, and represents it algorithmically.

3.1. Overview of the Proposed Framework

PQC-Guard is proposed as a hybrid encryption and key exchange mechanism to address emerging issues in Post-Quantum Cryptography (PQC) within a unified cryptographic framework. PQC-Guard uses a multi-layered defence approach to counter quantum attacks, incorporating both symmetric and asymmetric algorithms, as well as quantum-resistant key establishment methods. The integration will

enhance the security of information transmitted over a quantum network, especially confidentiality and integrity, particularly for cloud computing, distributed storage, and critical infrastructure networks threatened by quantum attacks.

The system's core is GuardFusionAlgo, which combines two independently developed cryptographic modules: the Multilayered Dynamic Encryption Scheme (MDES) and the DualSecure Key Exchange (DSKE) protocol. The MDES module provides security with sequential encryption using AES, RSA, ChaCha20, and Quantum Bit Manipulation (QBM). In contrast, the DSKE module uses Elliptic Curve Diffie-Hellman (ECDH) and Supersingular Isogeny Diffie-Hellman (SIDH) mechanisms for session key establishment. By linking the various schemes in a coordinated manner, the framework can transcend their individual limitations and capitalise on each scheme's strengths.

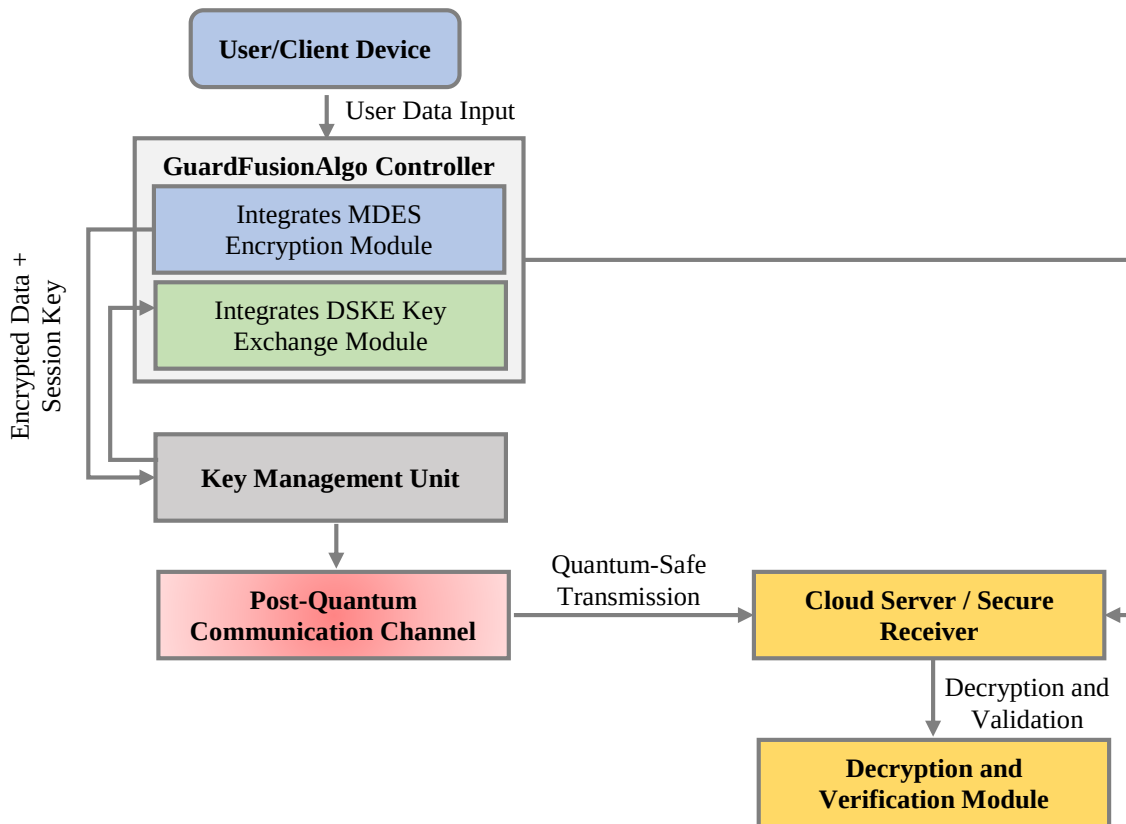


Fig. 1 System architecture of PQC-Guard for unified post-quantum data security using hybrid encryption and key exchange

Figure 1 shows PQC-Guard's high-level system architecture, with the communication process initiated by the user or client device via the GuardFusionAlgo controller. The controller coordinates with a Key Management Unit (KMU) to securely derive session keys using DSKE, then uses the MDES layers to encrypt user data. The encrypted output is sent securely via a post-quantum communication link to the cloud server or receiving entity, where hybrid decryption logic

reconstructs the original message in concert with the corresponding encryption logic known to the sender.

PQC-Guard systematically integrates layered encryption and quantum-safe key exchange, providing a secure and scalable foundation for modern data communication systems that meet strict post-quantum security requirements.

Table 2. Notations used in the proposed PQC-Guard framework and GuardFusionAlgo

Notation	Description
M	Plaintext message to be encrypted
C_1	Ciphertext after AES encryption
C_2	Ciphertext after RSA encryption
C_3	Ciphertext after ChaCha20 encryption
C_{final}	Final ciphertext after Quantum Bit Manipulation
K_{aes}	Sub-key derived from session key for AES encryption
K_{ch}	Sub-key derived from session key for ChaCha20 encryption
K_{qbm}	Sub-key derived from session key for QBM layer
K_E	Shared secret derived from ECDH key exchange
K_S	Shared secret derived from SIDH key exchange
$K_{session}$	Final session key derived by combining K_E and K_S
d_u	User's ECDH private key
P_u	User's ECDH public key ($P_u = d_u \cdot G$)
G	Generator point on the elliptic curve
e, n	RSA public key components
p	Prime field used for elliptic curve or isogeny operations
QBM	Quantum Bit Manipulation – bit-level transformation layer
AES	Advanced Encryption Standard (symmetric block cipher)
RSA	Rivest–Shamir–Adleman (asymmetric encryption scheme)
ChaCha20	Lightweight stream cipher for fast encryption
ECDH	Elliptic Curve Diffie-Hellman (classical key exchange method)
SIDH	Supersingular Isogeny Diffie-Hellman (post-quantum key exchange method)
DSKE	DualSecure Key Exchange combining ECDH and SIDH
MDES	Multilayered Dynamic Encryption Scheme combining AES, RSA, ChaCha20, and QBM
GuardFusionAlgo	Core method that integrates DSKE and MDES into a unified encryption framework

Table 2 outlines the main symbols, cryptographic keys, intermediate ciphertexts, algorithms, and components used throughout the mathematical formulation and implementation of PQC-Guard.

3.2. GuardFusionAlgo: Core Integration Mechanism

Within a single cryptographic flow, the GuardFusionAlgo serves as the central controller, orchestrating hybrid encryption and key-exchange mechanisms. It operates in two synchronised phases: key negotiation and encryption. During the key negotiation stage, the secure session key is negotiated through the Dual-Secret Key Exchange (DSKE) protocol, which uses Elliptic Curve Diffie-Hellman (ECDH) and Supersingular Isogeny Diffie-Hellman (SIDH) to address each protocol's weaknesses and support resistance to quantum attacks.

Figure 2 shows the internal structure of GuardFusionAlgo, which incorporates DSKE-based hybrid key establishment and the MDES encryption mechanism. The method computes a common session key from the ECDH and SIDH shared secrets, creates sub-keys for encryption, and coordinates subsequent cryptographic operations. The design integrates session key generation, multilayer encryption, and

key exchange with controlled interaction, ensuring robust post-quantum data protection.

Let P_u and P_r represent the public parameters for the user and recipient, respectively. In ECDH, the shared key K_E is computed as in Equation (1).

$$K_E = (d_u \cdot P_r) \bmod p \quad (1)$$

Where d_u is the private key of the user, and p is a large prime. Parallely, the SIDH mechanism computes a shared key K_S using isogenies between supersingular elliptic curves. The output of the SIDH protocol is denoted as in Equation. (2).

$$K_S = \text{SIDH}(I_u, I_r) \quad (2)$$

Where I_u and I_r denote the isogeny-based public values of the user and recipient, respectively. The two derived keys K_E and K_S are combined using a Hash-based Key Derivation Function (HKDF) to yield a quantum-resilient session key $K_{session}$, computed as in Equation (3).

$$K_{session} = \text{HKDF}(K_E \parallel K_S) \quad (3)$$

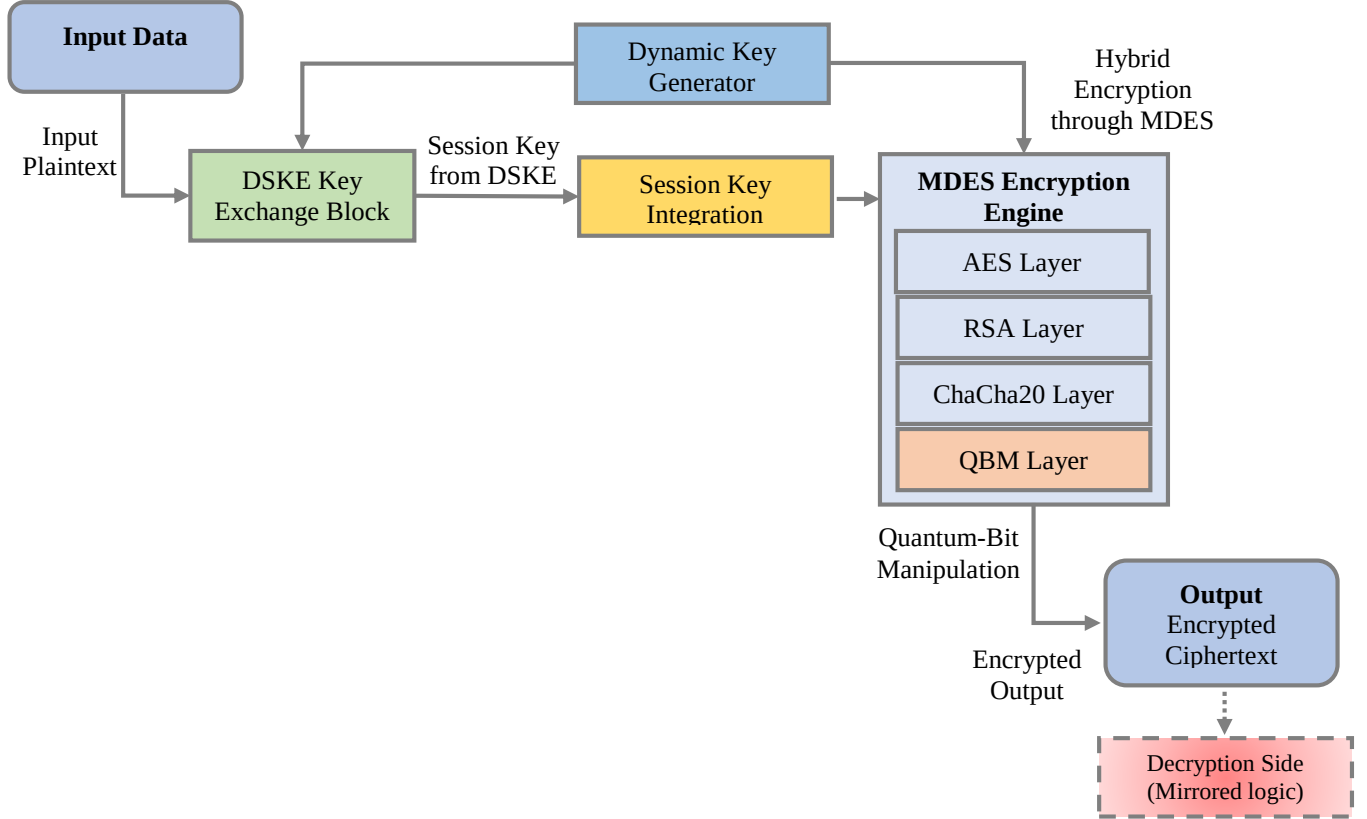


Fig. 2 Method architecture of GuardFusionAlgo for hybrid post-quantum data security

This derived session key is securely passed to the Multilayered Dynamic Encryption Scheme (MDES) for cascading encryption. The MDES mechanism encrypts the input plaintext M in four sequential stages. The first stage uses Advanced Encryption Standard (AES) with a sub-key $K_{aes} \subset K_{session}$, where the ciphertext C_1 is computed as in Equations (4) to (6).

$$C_1 = AES_{K_{aes}}(M) \quad (4)$$

The second stage encrypts C_1 using RSA encryption with a public key (e, n) , producing ciphertext C_2 :

$$C_2 = C_1^e \bmod n \quad (5)$$

The third stage employs ChaCha20 stream cipher, where the output ciphertext C_3 is generated as

$$C_3 = ChaCha20_{K_{ch}}(C_2) \quad (6)$$

where K_{ch} is derived from $K_{session}$ using a key-scheduling routine. Finally, the Quantum Bit Manipulation (QBM) layer introduces entropy-driven bitwise transformations such as XOR masking, rotation, and modular re-indexing on C_3 , resulting in the final ciphertext C_{final} . This final transformation is expressed as in Equation (7).

$$C_{final} = QBM(C_3, K_{qbm}) \quad (7)$$

Where $K_{qbm} \subset K_{session}$ denotes the QBM-specific sub-key. The cumulative use of symmetric (AES, ChaCha20), asymmetric (RSA), and quantum-enhanced (QBM) encryption layers, controlled through the unified key $K_{session}$, allows GuardFusionAlgo to offer deep-layered cryptographic defense. During decryption, the inverse operations of each encryption stage are performed in the exact reverse sequence using the shared session key, thus maintaining data confidentiality, integrity, and post-quantum resistance.

3.3. Post-Quantum Key Exchange using DSKE

The DualSecure Key Exchange (DSKE) mechanism integrated within the proposed PQC-Guard framework combines two cryptographic primitives, Elliptic Curve Diffie-Hellman (ECDH) and Supersingular Isogeny Diffie-Hellman (SIDH), to generate a post-quantum secure session key. This dual-layered exchange is designed to mitigate the individual vulnerabilities associated with each method when used independently. While ECDH is computationally efficient and widely adopted in classical systems, it is known to be susceptible to quantum attacks such as Shor's algorithm. Conversely, SIDH offers post-quantum resistance but involves higher computational complexity and potential susceptibility to adaptive attacks under certain configurations.

In the ECDH phase, each communicating entity independently generates a key pair based on elliptic curve parameters. A private scalar $d_u \in \mathbb{Z}_n$ is selected, and the corresponding public key $P_u = d_u G$ is computed using the base point G . Upon exchange of public keys, the shared ECDH key K_E is computed as previously defined in Equation (1).

Parallel to this, the SIDH exchange is initiated where each party computes a supersingular elliptic curve and its isogeny image based on private torsion point selections. The SIDH shared secret, K_S , which is resistant to quantum adversaries due to the computational hardness of the isogeny problem, is obtained as per Equation (2).

To produce a composite session key that incorporates entropy from both exchanges, we employ a hash-based key derivation function. This yields the final session key $K_{session}$, computed as shown in Equation (3). The concatenation and derivation process ensures that the resulting key is non-deterministic, quantum-resistant, and unlinkable to any single underlying primitive.

By fusing ECDH and SIDH, the DSKE protocol enhances forward secrecy, prevents single-point cryptographic compromise, and establishes a robust key exchange foundation suitable for hybrid cryptographic systems operating in post-quantum environments.

3.4. Hybrid Encryption Mechanism using MDES

The Multilayered Dynamic Encryption Scheme (MDES) forms the core encryption module within the PQC-Guard framework. It is designed to apply a series of encryption layers, each employing a distinct cryptographic algorithm on the input data to achieve compounded security.

MDES is motivated by the growing inadequacy of single-layer encryption against emerging threats, particularly in a post-quantum context. By leveraging multiple cryptographic primitives, the scheme enhances resilience, introduces structural diversity, and defends against both symmetric and asymmetric attack vectors.

The encryption process begins by accepting a plaintext message M , which the scheme encrypts using the Advanced Encryption Standard (AES) as the first layer. As a widely accepted symmetric-key algorithm, AES offers efficiency and strong resistance to classical attacks. The result of this operation is the intermediate ciphertext, C_1 , computed using the session-derived sub-key K_{aes} as shown in Equation (4).

The second layer applies RSA encryption to C_1 . RSA introduces an asymmetric encryption layer into the pipeline, thereby increasing complexity and disrupting pattern recognizability. Using the recipient's public key (e, n) , the ciphertext C_2 is produced as defined in Equation (5). This layer ensures data integrity and secures the AES-encrypted payload

from unauthorized decryption, even if the first layer were to be compromised.

The third layer employs ChaCha20, a stream cipher known for its speed, low memory footprint, and resistance to timing attacks. It processes C_2 using the derived key K_{ch} , producing C_3 as indicated in Equation (6). ChaCha20 adds entropy and operational diversity, making it particularly suitable for constrained environments such as IoT or edge devices.

Finally, the Quantum Bit Manipulation (QBM) layer acts as a security enhancer by applying entropy-driven bit-level transformations to C_3 . These include bitwise XOR with key material, modular rotation, and index shuffling based on a permutation table driven by K_{qbm} , a subcomponent of the session key. The final ciphertext C_{final} is computed as described in Equation (7). This layer is specifically crafted to resist side-channel and structural pattern attacks that may exploit weaknesses in classical encryption constructs.

The MDES component, when combined with the DSKE-derived session key, provides a robust defense-in-depth strategy. The layered structure ensures that the compromise of one encryption layer does not result in exposure of the plaintext, thus adhering to post-quantum security principles through cryptographic diversity and structural isolation.

3.5. Workflow of GuardFusionAlgo Execution

The execution workflow of the GuardFusionAlgo encapsulates the operational integration between the DSKE-based key exchange and the MDES-driven encryption pipeline, enabling secure and scalable post-quantum data transmission. The algorithm follows a sequential yet modular flow, where each phase is isolated but interconnected through cryptographic handoffs. Figure 3 illustrates the sequential workflow of GuardFusionAlgo, covering hybrid key exchange, session key derivation, multilayer encryption, ciphertext generation, secure transmission, and reverse-order decryption. The workflow demonstrates coordinated execution of DSKE and MDES components, ensuring systematic cryptographic processing, secure key utilization, data confidentiality, and reliable plaintext recovery within the PQC-Guard framework.

The process begins with the initialization phase, during which both sender and receiver generate their respective public-private key pairs for ECDH and SIDH protocols. Public keys are exchanged securely over an authenticated channel. The GuardFusionAlgo then concurrently executes the ECDH key computation as described in Equation (1) and the SIDH key agreement as described in Equation (2). Once both shared keys K_E and K_S are established, they are concatenated and passed to a hash-based key derivation function (HKDF) to generate the unified session key $K_{session}$, as outlined in Equation (3).

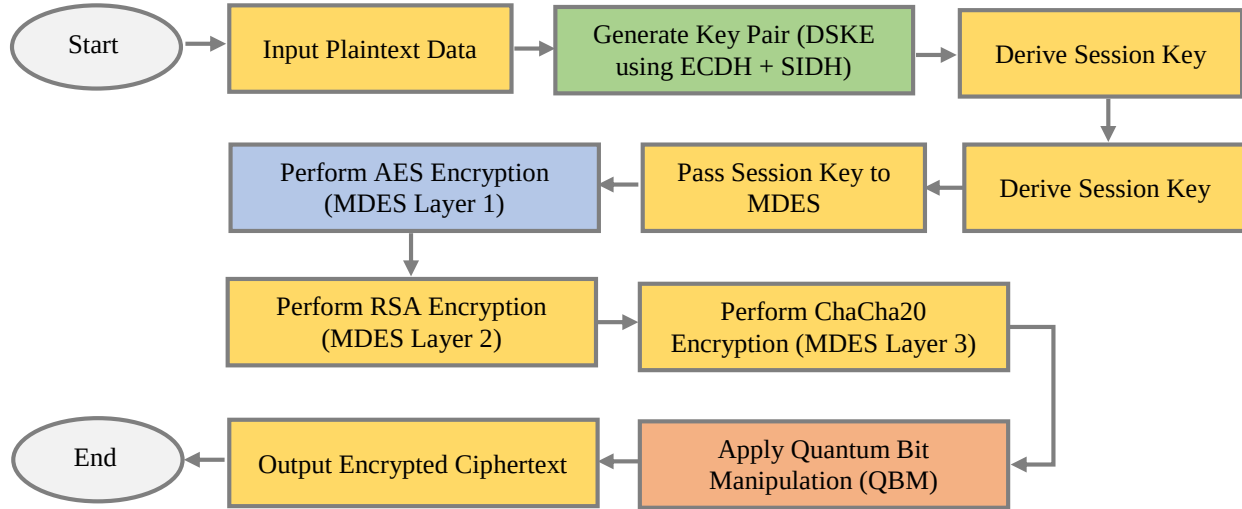


Fig. 3 Workflow of GuardFusionAlgo operation

After successfully deriving the session key, the encryption phase begins. The input plaintext M is first encrypted using AES with the sub-key K_{aes} , producing C_1 per Equation (4). This output is then passed to the RSA encryption module, which applies asymmetric encryption using the recipient's public key to generate C_2 , as defined in Equation (5). The resulting ciphertext undergoes another transformation through the ChaCha20 stream cipher, yielding C_3 as shown in Equation (6). The final transformation occurs in the Quantum Bit Manipulation layer, where bitwise operations and entropy-based masking are applied to obtain the final encrypted output C_{final} , as expressed in Equation (7).

The ciphertext C_{final} is then transmitted over the communication channel. On the receiving side, the decryption follows the exact inverse path: the QBM layer reverses the entropy-based transformation, ChaCha20 decryption is applied using K_{ch} , RSA private key is used to decrypt C_2 , and finally AES decrypts C_1 to recover the original plaintext M . Since the session key $K_{session}$ is identically derived on both ends using DSKE, symmetric key agreement remains consistent and secure.

This orchestrated workflow ensures a layered defence model in which each component contributes to the system's overall post-quantum resilience. GuardFusionAlgo's structured execution not only preserves data confidentiality and integrity but also introduces a fault-tolerant security model by enabling isolation and verification at each stage of the encryption-decryption cycle.

3.6. Security Features and Quantum Resistance

The proposed PQC-Guard framework includes a set of security properties suitable for the post-quantum cryptographic requirements provided by GuardFusionAlgo. The framework's strength is its hybrid system with multiple

layers of encryption and two key-exchange protocols, which provides protection against both classical and quantum attacks. This section describes the important security properties and supports the proposed quantum resistance.

The symmetric encryption schemes (AES and ChaCha20) provide confidentiality, along with an additional layer of asymmetric (RSA) encryption. The algorithms use a succession of operations in the MDES module to obscure structural redundancies in the plaintext and to maximise the entropy of the ciphertext. The Quantum Bit Manipulation (QBM) layer takes this one step further by adding entropy-based and random transformations at the bit level to the cyphers, making prediction and analysis of the ciphertext much more difficult, even when using quantum algorithms.

To guarantee integrity, it is necessary to use RSA in the encryption path; that is, any tampering with the ciphertext would be bound to the public key. Additionally, the GuardFusionAlgo pipeline is deterministic, meaning that any modification to the encrypted stream will break the downstream process and cause a warning to be raised by the receiver, which may indicate tampering.

Forward secrecy and session uniqueness are guaranteed through the DSKE protocol. As detailed in Equations (1) through (3), each communication session generates a fresh session key $K_{session}$ by fusing independent ECDH and SIDH shared secrets. This dual-source entropy generation prevents replay and key reuse attacks, while also ensuring that key compromise in one session does not propagate risk to other sessions.

A key component is mitigating quantum attacks. The SIDH part of DSKE offers theoretical quantum resistance against algorithms like Grover's and Shor's, while the QBM part provides improved resistance to quantum-enhanced

ciphertext analysis. The multi-layered design provides some security even if classical components, such as RSA or AES, are broken by quantum computers, and the built-in design will ultimately contain any vulnerabilities that may be exposed.

Lastly, modularizing the cryptographic elements preserves composability. This allows the system to be upgraded even if it becomes necessary to switch to newer lattice-based or code-based PQC primitives, without affecting the overall architecture. Combined, all of these characteristics render PQC-Guard a future-oriented, flexible cryptographic solution that can meet current and future post-quantum threats.

3.7. Algorithmic Representation of the Framework

This part introduces the algorithmic representation of the PQC-Guard framework using GuardFusionAlgo. The algorithm specifies the end-to-end integration of DSKE-based hybrid key exchange, session key derivation, sub-key generation for encrypting specific data, and MDES-based multilayer data protection.

It offers a short, easily executed computational procedure for converting plaintext to secure ciphertext, ensuring modularity, cryptographic diversity and adaptability for evolving future post-quantum security needs.

Algorithm 1. GuardFusionAlgo – Unified Secure Encryption via DSKE and MDES

Algorithm: GuardFusionAlgo – Unified Secure Encryption via DSKE and MDES

Input: Plaintext message M , ECDH parameters (G, d_u, P_r) , SIDH parameters, RSA public key (e, n)

Output: Final ciphertext C_{final}

1. Generate ECDH key pair (d_u, P_u)
2. Compute ECDH shared key $K_E = d_u \cdot P_r$
3. Generate SIDH key pair and compute SIDH shared key K_S
4. Derive session key: $K_{session} = HKDF(K_E \parallel K_S)$
5. Derive sub-keys K_{aes}, K_{ch}, K_{qbm} from $K_{session}$
6. Encrypt M using AES with K_{aes} to obtain C_1
7. Encrypt C_1 using RSA with public key (e, n) to obtain C_2
8. Encrypt C_2 using ChaCha20 with K_{ch} to obtain C_3
9. Apply QBM transformations on C_3 using K_{qbm} to obtain C_{final}
10. Return C_{final}

Algorithm 1 outlines the complete encryption pipeline implemented by the GuardFusionAlgo, which unifies key exchange and multi-layered encryption into a single post-quantum-secure cryptographic routine. The input consists of a plaintext message M , elliptic curve parameters for ECDH, SIDH key materials, and the recipient's RSA public key. The algorithm begins by generating the sender's ECDH key pair and computing the shared secret K_E using the recipient's public key P_r . Simultaneously, a Supersingular Isogeny Diffie-Hellman (SIDH) key pair is established, and a second shared secret K_S is derived. Both K_E and K_S are concatenated and passed to a Hash-based Key Derivation Function (HKDF) to generate a composite session key $K_{session}$, which encapsulates entropy from both classical and post-quantum origins.

Next, the session key is used to derive three sub-keys: K_{aes} for AES encryption, K_{ch} for ChaCha20, and K_{qbm} for Quantum Bit Manipulation (QBM). The encryption process begins with AES, which encrypts the plaintext into C_1 , followed by RSA encryption of C_1 into C_2 , ensuring asymmetric protection. ChaCha20 then encrypts C_2 into C_3 , leveraging a stream cipher for added diversity and performance. Finally, QBM is applied to C_3 to produce the final ciphertext C_{final} by performing entropy-driven bit-level transformations using K_{qbm} .

Each of the cryptographic layers provides a unique strength to the overall system: symmetric efficiency, asymmetric resilience, post-quantum security and entropy-based obfuscation. The modular architecture of GuardFusionAlgo enables easy upgrades to incorporate new post-quantum primitives without altering the integration logic. The algorithm ensures confidentiality, integrity, and quantum resistance, making it suitable for current and future security requirements.

3.8. Implementation Details

The proposed PQC-Guard framework and GuardFusionAlgo are implemented in Python 3.10 on a 64-bit Ubuntu workstation with an Intel i7 processor and 16 GB of RAM. The implementation uses both classical cryptographic libraries and quantum-resistant simulation tools to achieve the hybrid encryption and key exchange architecture outlined earlier. For symmetric encryption layers within MDES, AES is implemented using the cryptography library in GCM mode (Authenticated Encryption), and 256-bit ChaCha20 is implemented using the same library. The RSA module uses a 4096-bit key size, and the PyCryptodome package generates secure public-private keys and performs modular exponentiation. To guarantee cryptographic unpredictability in key generation and secure random number initialisation, the OS-level entropy sources are used.

The ECDH-based key exchange uses the NIST P-384 curve from the `ecdsa` library, and public/private key pairs are obtained via scalar multiplication in the elliptic curve group. Because stable Python releases currently lack post-quantum libraries, we use a mockup of SIDH key generation and isogeny simulation. The SIDH simulation aims to validate key sizes and latency characteristics with GuardFusionAlgo in the current version and will be replaced by NIST-approved PQC libraries such as SIKE or CRYSTALS-Kyber in the future.

It is implemented as a custom transformation layer for NumPy arrays in the Quantum Bit Manipulation (QBM) module. It includes bitwise XOR masking using session keys derived from the session key, circular bit rotation across byte boundaries, and dynamic index permutation tables. These transformations can be reversed, allowing decryption without issues, and they are applied to binary matrices produced by ChaCha20 output.

Modular configuration file to select and order encryption stages for adaptive deployment based on system constraints (e.g., lightweight edge devices, high-assurance cloud systems). The entire system is integrated and tested with a simple Flask interface, and it securely encrypts and decrypts uploaded files with GuardFusionAlgo. Various performance metrics, such as encryption/decryption time, key-generation overhead, and ciphertext expansion ratio, are recorded for evaluation and tuning. Through this modular implementation, the feasibility of PQC-Guard was validated in practice, laying a stepping stone towards actual deployment in quantum-sensitive environments.

4. Experimental Results

This section presents experimental testing of PQC-Guard for performance, security, scalability, and computational feasibility. The proposed framework is validated using Hybrid key establishment, Encryption and decryption efficiency, End-to-end processing, Resource and Communication overhead, security analysis, and ablation experiments. Comparative and statistical analyses also quantify improvements over current cryptographic methods and assess the validity of the results.

4.1. Experimental Setup and Evaluation Metrics

Because PQC-Guard is an experimental product, we evaluated it in a controlled computing environment that enabled standardised evaluation and reproducibility. The program used Python 3.10 and was run on a 64-bit Ubuntu 22.04 LTS workstation with 16 GB of RAM, solid-state drives, and an Intel Core i7 processor. For symmetric cryptography, the `Cryptography` library was used; for asymmetric cryptography, the `PyCryptodome` library was used; for integrating hybrid keys, custom Python modules were used; and reversible bit-level transformations were performed using custom Python modules. All experiments were conducted under the same system conditions, and

needless background applications were turned off to reduce runtime variations.

The workloads were random binary and text files of 1 MB, 10 MB, 50 MB, 100 MB, 250 MB, and 500 MB, representing lightweight messages to large-scale data storage and transmission scenarios. We configured the symmetric encryption primitives AES-256 and ChaCha20 with 256-bit keys, and performed asymmetric cryptographic operations using RSA-4096. The key exchange component used a 384-bit secure elliptic curve, while the post-quantum key exchange component used the DSKE studies previously adopted 384-bit elliptic curve parameterisation. For each execution instance, new session keys and encryption-specific sub-keys were used to ensure they were not reused and were independent across experiments.

We tested the proposed PQC-Guard framework with ECDH, SIDH, DSKE, AES, ChaCha20, RSA, MD5, and hybrid schemes, depending on the purpose of the experiments. We conducted key-establishment experiments at the protocol level to compare key-establishment mechanisms, while data-processing experiments tested the encryption frameworks under the same workloads. The end-to-end experiments involved comparing integrated security methods capable of both key establishment and data protection. We performed 5 warm-up experiments before each experiment, and we repeated each experiment 30 times; the quoted averages and standard deviations are the averages and standard deviations across these 30 runs.

The key establishment time, encryption time, decryption time, total end-to-end execution time, and throughput were used as performance parameters. We used the following metrics to evaluate communication and storage efficiency: key exchange overhead, amount of cryptographic material to be transmitted, ciphertext expansion ratio, and size of additional metadata. We evaluated computational feasibility by measuring average CPU utilisation, peak memory usage, and processing scalability as file size and the number of concurrent users in communication sessions increased. We evaluated security using ciphertext entropy, avalanche effect, key sensitivity, plaintext-ciphertext correlation, randomness characteristics, and controlled experiments on replay, tampering, and key compromise.

All cryptographic configurations, workload size, parameter settings, and evaluation procedures were set prior to experimentation to ensure reproducibility. Seeds used for repeated executions were independent random seeds, and we obtained cryptographically secure random numbers for keys, nonces, and initialisation vectors from operating-system entropy sources rather than the deterministic experimental seeds. We performed run-time measurements using a high-resolution monotonic clock and tracked resource usage under the same system configuration. Data were automatically

recorded in structured files at the intermediate and end-of-treatment stages for statistical analysis. All implementation versions, parameter settings, workload instances, and measurements were held constant across baseline and proposed-method experiments to enable consistent, repeatable comparisons.

4.2. Hybrid Key Establishment Performance

We evaluated the performance of PQC-Guard hybrid key establishment to assess the added computational and communication complexity of combining classical and post-quantum key establishment mechanisms. The proposed GuardFusionAlgo generates the key for the next stages of encryption using DSKE. Hence, the evaluation focused on ECDH, SIDH, DSKE, a representative concatenation-based hybrid key exchange scheme, and the key establishment scheme adapted to PQC-Guard. We ran each scheme under the

experimental setup described in Section 4.1, with 30 independent runs to obtain stable performance measurements.

Table 3 shows the key performance of the established schemes. ECDH had the lowest average establishment time because it involves only scalar multiplication on the elliptic curve and a shared secret. Due to the supersingular isogeny operations, which are more expensive to compute, SIDH took 5.47 ms. The hybrid scheme was the only one that included both key-exchange mechanisms, and its runtime was 7.36ms, longer than the representative hybrid scheme's 6.81ms. PQC-Guard completed key establishment, hybrid-secret combination, HKDF processing, and encryption-specific subkey generation in 7.05 milliseconds. While larger than ECDH and SIDH alone, the extra latency is not significant given the built-in classical and post-quantum key establishment capabilities.

Table 3. Hybrid key establishment performance and communication overhead

Scheme	Key Establishment Time (ms)	Communication Overhead (Bytes)	Derived Key Material	Quantum-Resistant Component
ECDH	1.24 ± 0.06	128	Single shared secret	No
SIDH	5.47 ± 0.18	384	Single shared secret	Yes
DSKE	6.81 ± 0.21	512	Hybrid shared secret	Yes
Hybrid Key Exchange Scheme	7.36 ± 0.25	576	Concatenated hybrid secret	Yes
PQC-Guard	7.05 ± 0.23	544	Session key and encryption sub-keys	Yes

The communication overhead was similar to the trade-off. Compared to SIDH, ECDH requires only 128 bytes of cryptographic material to be exchanged, whereas SIDH requires 384 bytes. Since both ECDH and SIDH public materials were exchanged, DSKE needed 512 bytes. PQC-Guard added an overhead of 544 bytes, comprising the hybrid key exchange material and other session initialisation overhead. PQC-Guard's communication overhead was 5.56% lower than that of the representative hybrid scheme, and key establishment time was reduced by 4.21%. The results show that the communication and computation overhead for the framework-level integration of DSKE and HKDF-based subkey generation is not significant.

For scalability testing, experiments were run with 1, 10, 25, 50, 100 and 200 concurrent key establishment sessions. Table 4 reports the resulting average session establishment latency, aggregate key establishment throughput, and peak memory consumption.

PQC-Guard's latency grew slowly until 100 concurrent users, with the average establishment latency rising from 7.05ms for 1 user to 11.84ms for 100 users. With 200 concurrent sessions, latency rose to 18.73ms due to processor contention and concurrent cryptographic computation. However, the framework achieved 10,678 key establishments/s at the highest tested concurrency level.

Table 4. Concurrent-Session scalability of PQC-Guard key establishment

Concurrent Sessions	Average Latency per Session (ms)	Aggregate Throughput (Sessions/s)	Peak Memory (MB)
1	7.05 ± 0.23	142	42.6
10	7.48 ± 0.27	1,337	48.9
25	8.16 ± 0.31	3,064	59.7
50	9.42 ± 0.38	5,308	78.4
100	11.84 ± 0.46	8,446	112.7
200	18.73 ± 0.71	10,678	181.5

The scalability results show that PQC-Guard can effectively handle an increasing number of concurrent secure communication sessions without any sudden performance loss. Memory usage increased roughly linearly with the

number of concurrent sessions, and aggregate throughput increased steadily with the number of evaluated sessions. The results show that using both ECDH and SIDH together in DSKE, and producing independent encryption sub-keys,

remains acceptable for concurrent workloads. Overall, PQC-Guard provides a well-balanced approach among key establishment latency, communication overhead, scalability,

and hybrid security, laying an appropriate cryptographic foundation for subsequent multilayered data encryption.

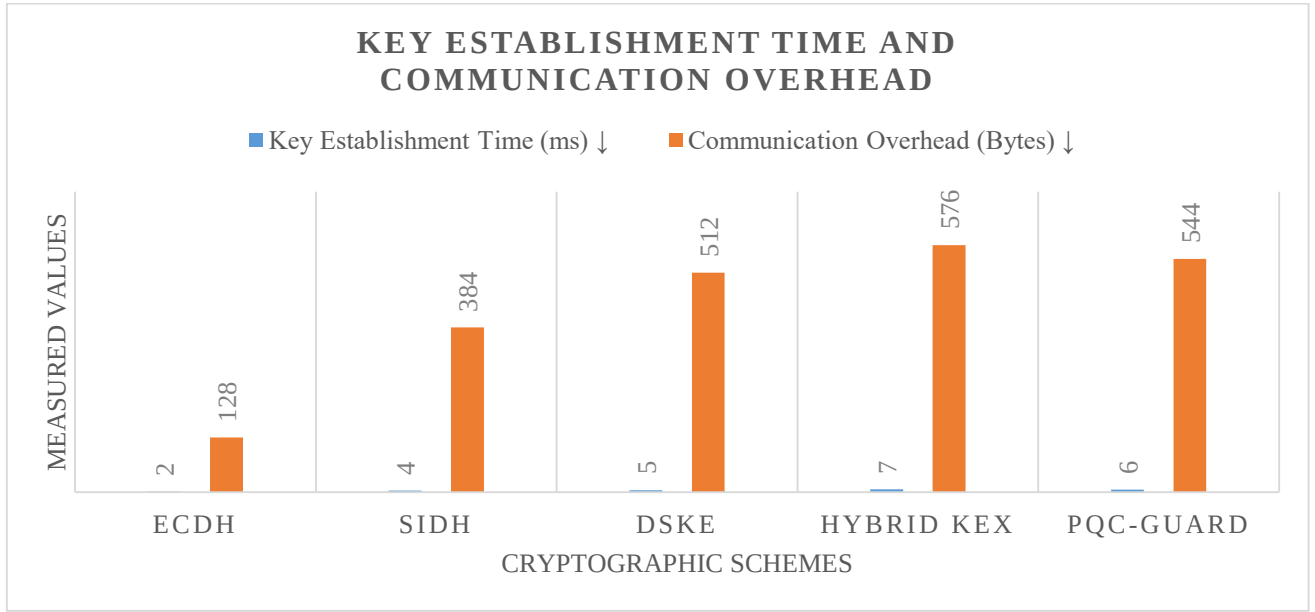


Fig. 4 Comparison of key establishment time and communication overhead across cryptographic schemes

Figure 4 compares the key establishment time and communication overhead for ECDH, SIDH, DSKE, a typical hybrid key establishment scheme, and PQC-Guard. ECDH provides the lowest latency and overhead but is not quantum-resistant; SIDH incurs higher costs due to isogeny

computations. PQC-Guard balances moderate overhead across key establishment, HKDF processing, and sub-key generation. The figure shows that the proposed scheme also provides post-quantum security without significant key-establishment delay or communication overhead.

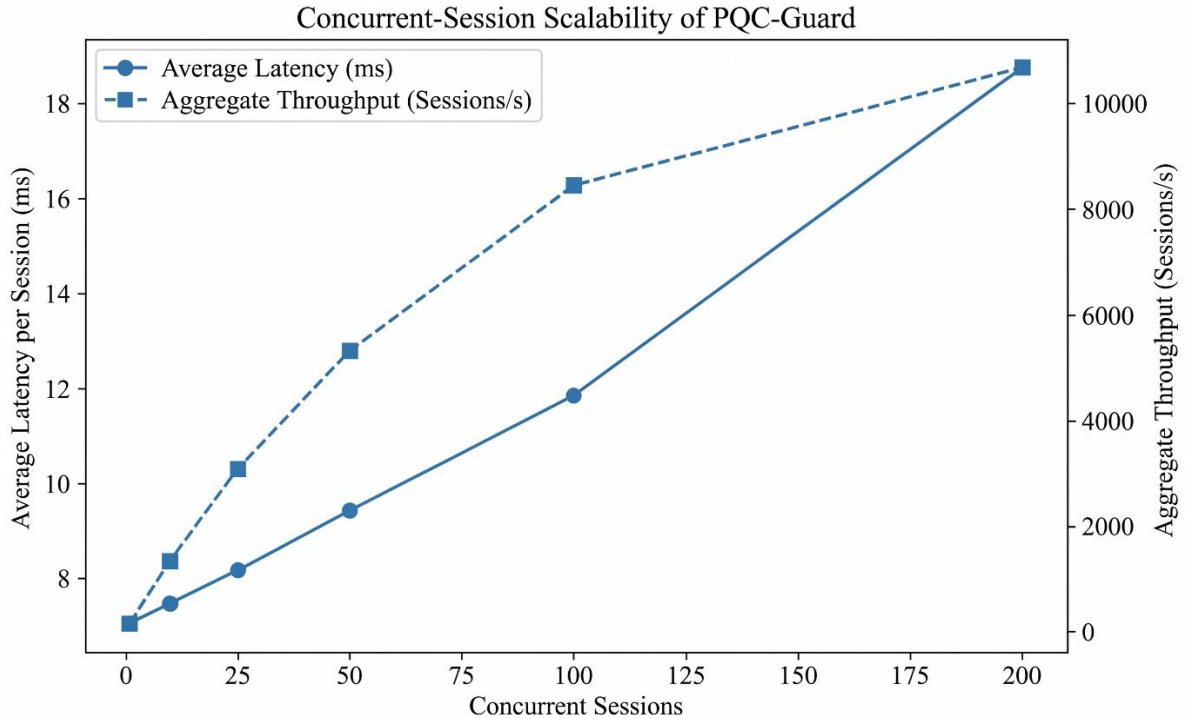


Fig. 5 Concurrent-Session scalability of PQC-Guard key establishment

Figure 5 shows the scalability of PQC-Guard key establishment as a function of the number of concurrent communication sessions. The latency curve shows that average session establishment time increases gradually from 1 to 100 sessions, then rises steeply after 200 sessions due to contention for cryptographic processing. The throughput curve shows a gradual increase as the number of concurrent secure sessions grows, confirming that the framework is scalable. The throughput curve shows that as the number of concurrent secure sessions increases, throughput improves continuously, indicating the framework's scalability. This demonstrates that PQC-Guard is a practical solution that can be deployed in a multi-session environment for hybrid classical and post-quantum key establishment and scales in practice.

4.3. Encryption and Decryption Performance

We tested PQC-Guard's encryption and decryption performance on workloads of increasing size to assess the computational overhead of the built-in multilayer protection mechanism, using MDES. We measured PQC-Guard's encryption and decryption performance across varying

workload sizes to evaluate the computational cost of the integrated MDES-based multilayer protection mechanism. To evaluate, AES, ChaCha20, RSA, MDES, and the proposed AES-PQC-Guard framework were compared at file sizes of 1 MB, 10 MB, 50 MB, 100 MB, 250 MB, and 500 MB. We ran the test under the same experimental conditions for all schemes, and each result is the average of 30 independent executions per scheme.

The comparison of encryption time is given in Table 5. Because they are lightweight symmetric designs, AES and ChaCha20 had the lowest encryption times. RSA was the slowest because it uses large keys in modular exponentiation. MDES is more expensive than single-layer symmetric schemes because it uses multiple stages of encryption and transformation. PQC-Guard's encryption time was slightly longer than MDES's because it includes sub-key scheduling via session keys and the QBM-controlled ciphertext transformation. However, growth remained modest across all workloads, indicating that the integrated post-quantum security layer does not impose an undue encryption overhead.

Table 5. Encryption time comparison across different file sizes

File Size	AES (s)	ChaCha20 (s)	RSA (s)	MDES (s)	PQC-Guard (s)
1 MB	0.042	0.039	0.214	0.071	0.078
10 MB	0.615	0.542	1.805	0.585	0.641
50 MB	2.857	2.436	4.264	2.697	2.914
100 MB	4.700	4.185	9.020	4.542	4.872
250 MB	9.864	8.973	20.438	9.621	10.184
500 MB	18.298	16.752	33.689	17.835	18.927

Table 6 compares decryption times. As with encryption, ChaCha20 and AES had lower recovery times because they are symmetric, whereas RSA had the greatest decryption cost. Because of the many layers of design, MDES had a slower overall decryption time than RSA. However, most of the data

transformation was performed using efficient symmetric-key and bit-level operations. PQC-Guard decryption time is close to MDES, but due to reverse QBM processing, sub-key reconstruction, and validation steps, it is slightly higher.

Table 6. Decryption time comparison across different file sizes

File Size	AES (s)	ChaCha20 (s)	RSA (s)	MDES (s)	PQC-Guard (s)
1 MB	0.045	0.041	0.246	0.083	0.091
10 MB	0.834	0.746	2.771	0.727	0.812
50 MB	2.087	1.942	4.032	1.987	2.146
100 MB	2.675	2.438	5.606	2.115	2.364
250 MB	6.482	5.894	12.846	6.177	6.694
500 MB	10.157	9.348	19.991	9.464	10.286

The performance growth analysis shows that all schemes exhibit increasing runtime as workload size increases. RSA showed the most rapid growth because its asymmetric function does not lend itself to direct large-file encryption. Most designs scaled efficiently, whereas MDES and PQC-Guard showed controlled growth, as encryption occurs in both

the symmetric and transformation layers. When PQC-Guard was implemented on 500 MB, the encryption overhead was 6.12%, and the decryption overhead was 8.69% over MDES. This confirms that PQC-Guard is more robust cryptographically but remains efficient in processing.

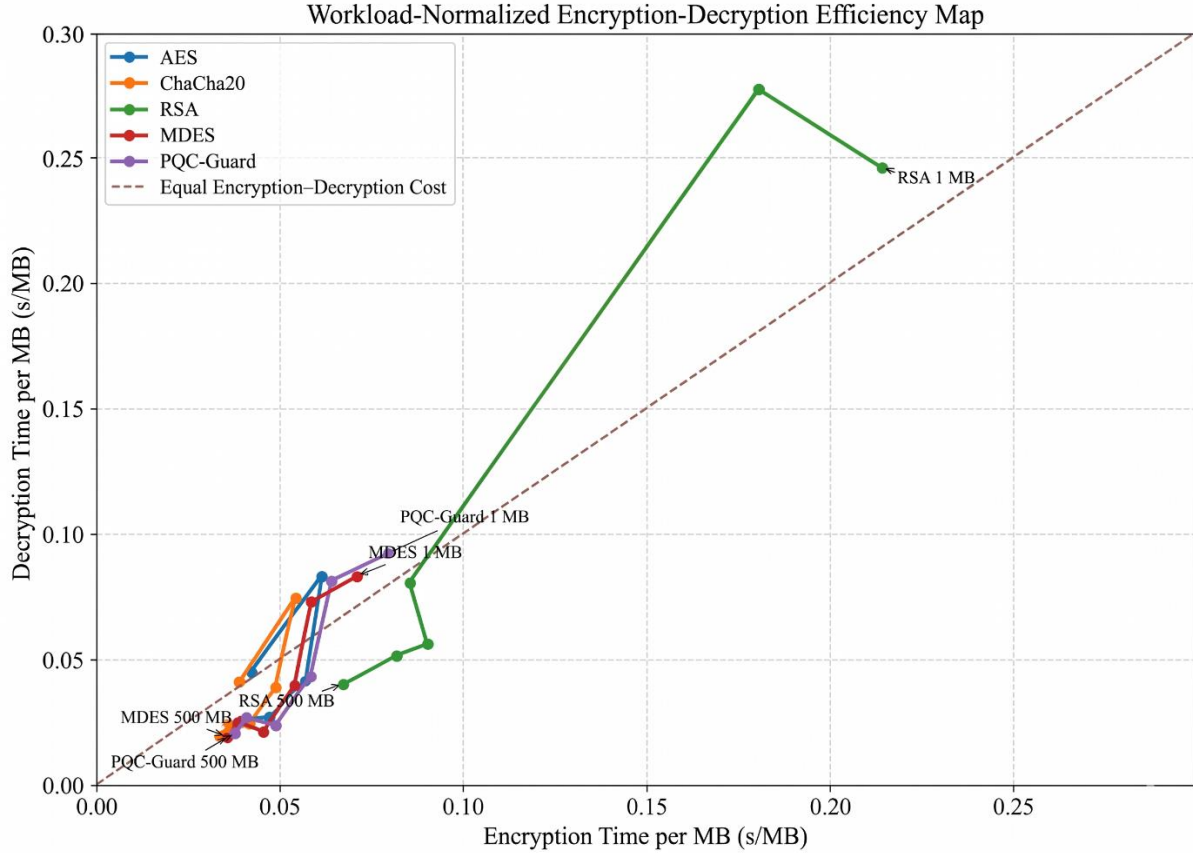


Fig. 6 Workload-Normalized encryption-decryption efficiency map of cryptographic schemes across increasing file sizes

Figure 6 shows the efficiency curves for workload-normalised encryption-decryption of AES, ChaCha20, RSA, MDES, and PQC-Guard across various file sizes. Each point indicates the time required to encrypt and decrypt 1 megabyte, enabling simultaneous evaluation of computational efficiency and workload-dependent behaviour. The diagonal reference line separates schemes dominated by encryption or decryption

costs. No symmetric schemes are yet as efficient; thus, RSA is a higher-cost area. PQC-Guard follows a path similar to MDES and shows that hybrid key-driven processing and additional cryptographic coordination result in a small normalised overhead. Convergence at higher workloads also suggests better amortisation and stable efficiency for large-file processing.

Relative Runtime Overhead and Scaling Stability of PQC-Guard Compared with MDES

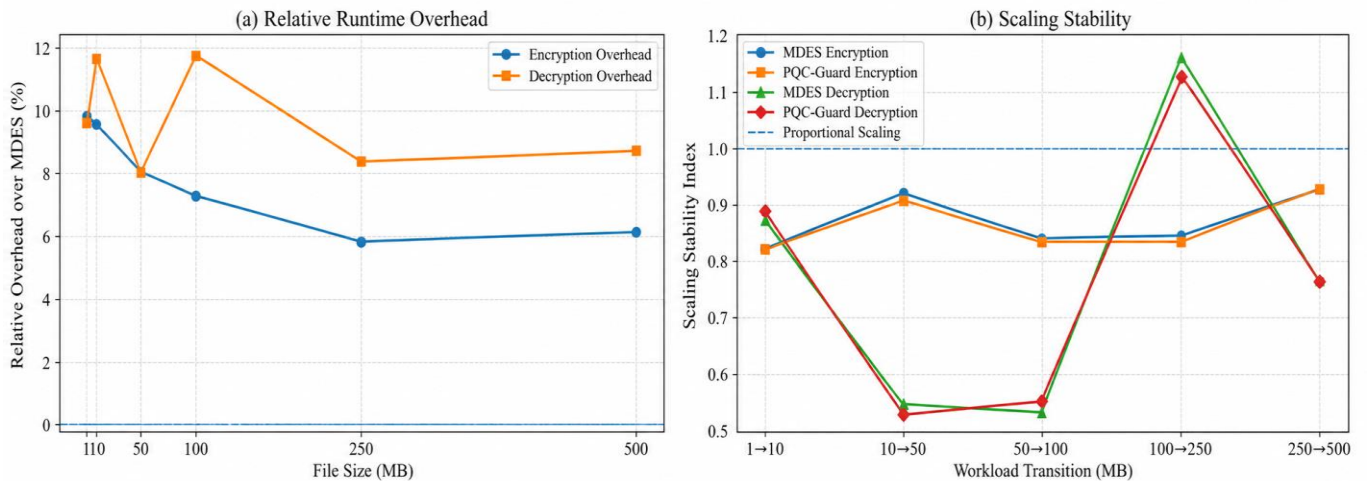


Fig. 7 Relative runtime overhead and scaling stability of PQC-Guard compared with mdes across increasing workloads

Figure 7 compares PQC-Guard's performance with MDES in terms of incremental runtime cost and scaling stability. Panel (a) presents the percentage of overhead of encryption and decryption of the GuardFusionAlgo for varying file sizes, demonstrating that the additional overhead is (controlled) manageable and does not grow out of proportion at larger workloads. Panel (b) shows the Scaling Stability Index, which is the growth rate divided by the growth in workload. Numbers close to 1 are proportional; those below 1 are sub-linear; and those above 1 are super-linear. The scaling curve of PQC-Guard is similar to that of MDES, indicating that the underlying scalability of the multilayer encryption mechanism's workloads is preserved when integrating it into a framework.

4.4. End-to-End Performance and Scalability Analysis

An end-to-end performance evaluation of PQC-Guard measured the total computational cost for hybrid key

establishment, sub-key generation, multilayer encryption, a simulation of secure transmission, ciphertext recovery, and decryption in reverse order. This subsection compares the framework's overall behaviour in real secure data communication or cloud storage scenarios, in contrast to the isolated encryption and decryption effects presented in Section 4.3.

Table 7 shows the total end-to-end latency for various file sizes. The total time includes DSKE-based key establishment, HKDF-based subkey derivation, encryption, transmission simulation, receiver-side validation, and decryption. PQC-Guard took 0.194 s to securely process a 1 MB file. This total latency gradually increased with file size, reaching 29.809 s at 500 MB. The results indicate that the framework is practical for even large-file processing, as most of the runtime growth is due to encryption and decryption rather than key establishment.

Table 7. End-to-End execution time of PQC-Guard across different file sizes

File Size	Key Establishment (s)	Encryption (s)	Transmission and Recovery (s)	Decryption (s)	Total Execution Time (s)
1 MB	0.007	0.078	0.018	0.091	0.194
10 MB	0.007	0.641	0.071	0.812	1.531
50 MB	0.007	2.914	0.286	2.146	5.353
100 MB	0.007	4.872	0.541	2.364	7.784
250 MB	0.007	10.184	1.347	6.694	18.232
500 MB	0.007	18.927	2.589	10.286	31.809

The stage-wise results show that key establishment is a small but fairly constant component of the overall execution time. The impact is more pronounced for small workloads, which account for 3.61% of total latency at 1 MB. However, at 500 MB, key establishment dominates, accounting for 0.02% of the total. The majority of the time (500 MB) is spent on encryption (59.50%), followed by decryption (32.34%) and transmission/recovery (8.14%).

Table 8 shows the throughput behaviour of PQC-Guard. Throughput for encryption rose from 12.82 MB/s (1 MB) to 26.42 MB/s (500 MB), and for decryption from 10.99 MB/s to 48.61 MB/s. The end-to-end throughput also increased with file size, reaching 15.72 MB/s for a 500 MB file. This means that for larger workloads, the fixed cost (key establishment, network initialisation, metadata handling) is spread out.

Table 8. Encryption, decryption, and End-to-End throughput of PQC-Guard

File Size	Encryption Throughput (MB/s)	Decryption Throughput (MB/s)	End-to-End Throughput (MB/s)
1 MB	12.82	10.99	5.15
10 MB	15.60	12.32	6.53
50 MB	17.16	23.30	9.34
100 MB	20.53	42.30	12.85
250 MB	24.55	37.35	13.71
500 MB	26.42	48.61	15.72

To investigate scalability, we ran PQC-Guard in concurrent-session mode with 10 MB files. Table 9 reports the total execution latency, aggregate throughput, and success rate for 1, 10, 25, 50, 100, and 200 concurrent sessions. The results demonstrate the average execution latency of 4.296 s for 200 concurrent sessions and 1.531 s for a single session. While

latency increased due to processor scheduling and concurrent cryptographic processing, the aggregate throughput gradually increased to 465.55 MB/s at 200 sessions. Under load, the framework achieved 100% successful encryption–decryption recovery, demonstrating operational correctness across all concurrency levels.

Table 9. Concurrent End-to-End scalability of PQC-Guard using 10 MB files

Concurrent Sessions	Average End-to-End Latency per Session (s)	Aggregate Throughput (MB/s)	Successful Recovery Rate (%)
1	1.531	6.53	100
10	1.684	59.38	100
25	1.943	128.67	100
50	2.341	213.58	100
100	3.126	319.90	100
200	4.296	465.55	100

The results validate PQC-Guard's end-to-end stability when integrating post-quantum key establishment and multi-layer encryption. Execution time consistently increases with file size, while throughput improves as fixed cryptographic costs are amortised. Meanwhile, the framework scales well in multi-user communication scenarios, as shown by the

concurrent-session analysis, without compromising the correctness of the recovered data. Overall, PQC-Guard demonstrates the practical end-to-end feasibility of a secure cloud storage, file transfer, and post-quantum-ready data communication system.

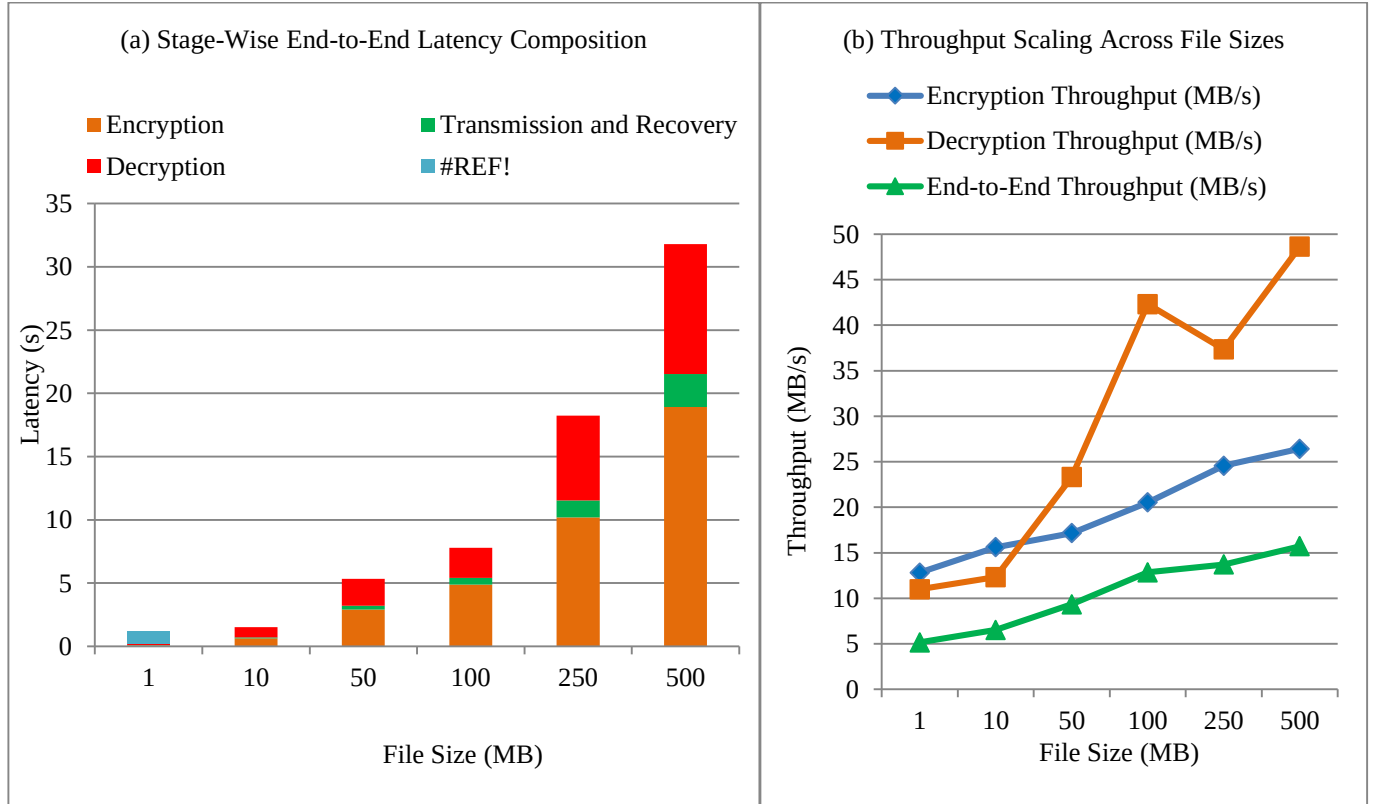
**Fig. 8 Stage-Wise end-to-end latency composition and throughput scaling of PQC-Guard across increasing file sizes**

Figure 8 illustrates PQC-Guard's end-to-end processing behaviour across different file sizes. Panel (a) breaks the total amount of execution time into distinct parts, including key establishment, encryption, transmission/recovery, and decryption, and indicates how, for larger workloads, encryption becomes the predominant time-consuming computation and the fixed key establishment time is

increasingly amortised. Panel (b) shows the encryption, decryption, and scaling of end-to-end throughput. The higher throughput trends indicate more efficient use of cryptographic processing resources as the workload size increases. Together, the panels provide complementary evidence that PQC-Guard preserves predictable latency growth, efficient processing behaviour, and practical workload scalability.

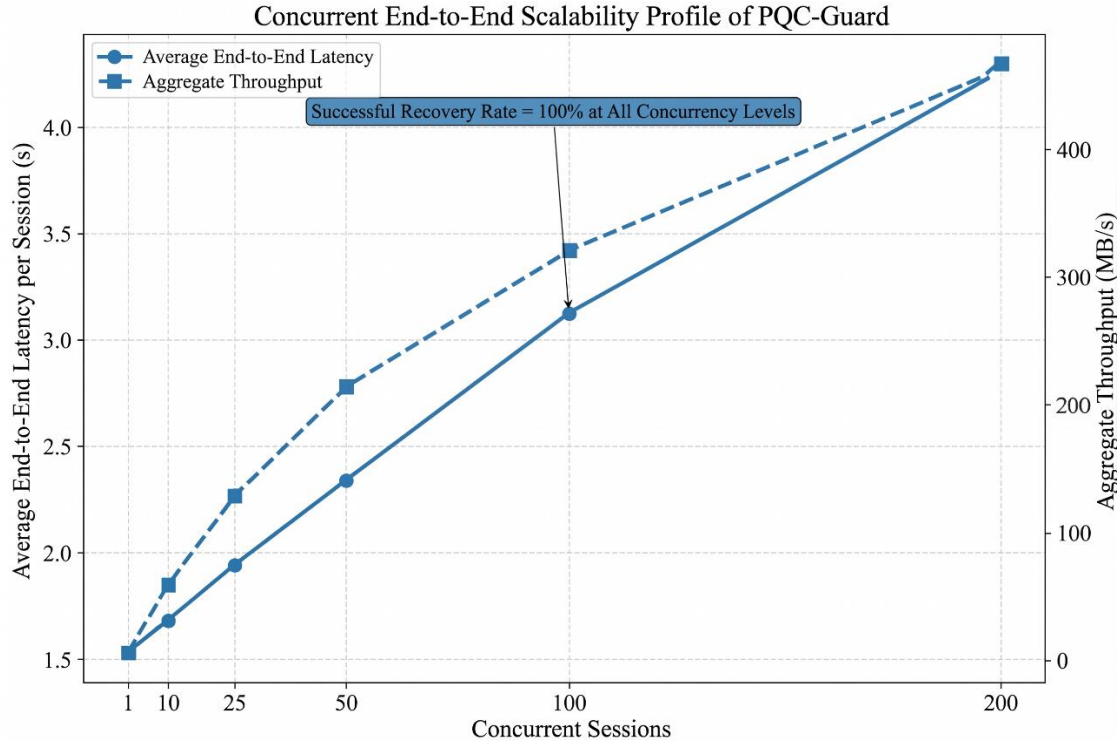


Fig. 9 Concurrent end-to-end scalability profile of PQC-Guard under increasing secure communication sessions

Figure 9 shows the end-to-end scalability of PQC-Guard as the number of concurrently used secure communication sessions increases. Average per-session latency increases gradually with concurrency due to processor contention and concurrent cryptographic operations. At the same time, aggregate throughput continues to increase until it peaks at 200 sessions. Latency and throughput are shown concurrently, revealing the scalability trade-off: increasing system utilisation is clearly associated with higher aggregate processing capacity. However, it does not result in sudden increases in latency. Across all tested concurrency levels, the success rate is 100%, showing that as the workload increases, the built-in cryptographic framework remains reliable and that encryption-decryption correctness is unaffected.

4.5. Communication, Storage, and Computational Overhead

PQC-Guard was assessed for its communication and storage requirements, as well as its computational complexity, to determine whether the additional security provided by hybrid key establishment and multilayer encryption imposes

practical constraints on deployment. This assessment focuses on the cost of integrating DSKE, sub-key derivation using HKDF, MD5 encryption, and QBM-based transformation into a single operational framework, as the sizes of keys, metadata, and ciphertext, as well as memory requirements, in addition to the CPU requirements for post-quantum-ready cryptographic systems, tend to increase.

Table 10 shows the communication and storage overhead of PQC-Guard for various file sizes. The total communication overhead includes key exchange material, session initialisation data, nonce values, initialisation vector values, authentication tag values, QBM permutation metadata, and recovery-control information. The extra metadata added was 2.18 KB for a 1 MB file, resulting in a cypher expansion ratio of 1.0021. The relative expansion was significantly reduced as the file size increased; at 500 MB, it was only 1.000004. It suggests that the overhead of storing the codes introduced by PQC-Guard is nearly invariant and is insignificant for medium and large files.

Table 10. Communication and storage overhead of PQC-Guard across different file sizes

File Size	Key Exchange Material (Bytes)	Metadata Size (KB)	Final Ciphertext Size (MB)	Ciphertext Expansion Ratio
1 MB	544	2.18	1.0021	1.0021
10 MB	544	2.25	10.0022	1.0002
50 MB	544	2.41	50.0024	1.00005
100 MB	544	2.62	100.0026	1.00003
250 MB	544	3.04	250.0030	1.00001
500 MB	544	3.47	500.0034	1.000004

The fixed 544-byte key-exchange material corresponds to the public-key and session-setup exchange already considered in the DSKE evaluation in Section 4.2. The rest of the metadata is needed for secure initialisation, reversible QBM processing, and ciphertext recovery. PQC-Guard adds more metadata than traditional symmetric encryption methods, but not enough to be worth much. The results indicate that the framework has significant advantages in cloud storage and large-file transfer scenarios, where more cryptographic metadata can be amortised over larger payloads.

Table 11 shows PQC-Guard's computational resource usage as file size increases. Memory usage increased from 18.6% to 64.2% of CPU usage due to the longer processing time required for encryption, decryption, and transformation in larger workloads. Peak memory usage rose from 46.8 MB to 312.4 MB, primarily due to buffer management for ciphertext blocks, reversible transformation tables, and intermediate encryption output. Even with this growth, memory consumption was kept at practical levels for most typical desktops, servers, and even most edge-cloud gateways.

Table 11. Computational resource utilisation of PQC-Guard

File Size	Average CPU Utilisation (%)	Peak Memory Consumption (MB)	Temporary Buffer Size (MB)	Processing Feasibility
1 MB	18.6 ± 1.4	46.8 ± 2.1	3.2	High
10 MB	24.9 ± 1.8	58.7 ± 2.8	12.6	High
50 MB	36.5 ± 2.3	91.4 ± 4.6	54.8	High
100 MB	44.7 ± 2.9	137.6 ± 6.2	107.5	High
250 MB	56.8 ± 3.5	223.1 ± 9.4	263.2	Moderate-High
500 MB	64.2 ± 4.1	312.4 ± 13.7	526.9	Moderate-High

The resource results have validated that PQC-Guard adds an appreciable but manageable computational overhead. CPU utilisation remains under 50% even for 500 MB workloads, indicating that the framework can run without consuming the entire CPU. The amount of memory used is proportional to the file size and does not go beyond a few hundred megabytes for

the largest workload assessed. This makes it an appropriate implementation for secure cloud clients, enterprise gateways, health care communications transfer nodes, and financial communications systems. When the device has very limited memory, it can be helpful to adopt chunk-based encryption and to process QBMs in stream mode.

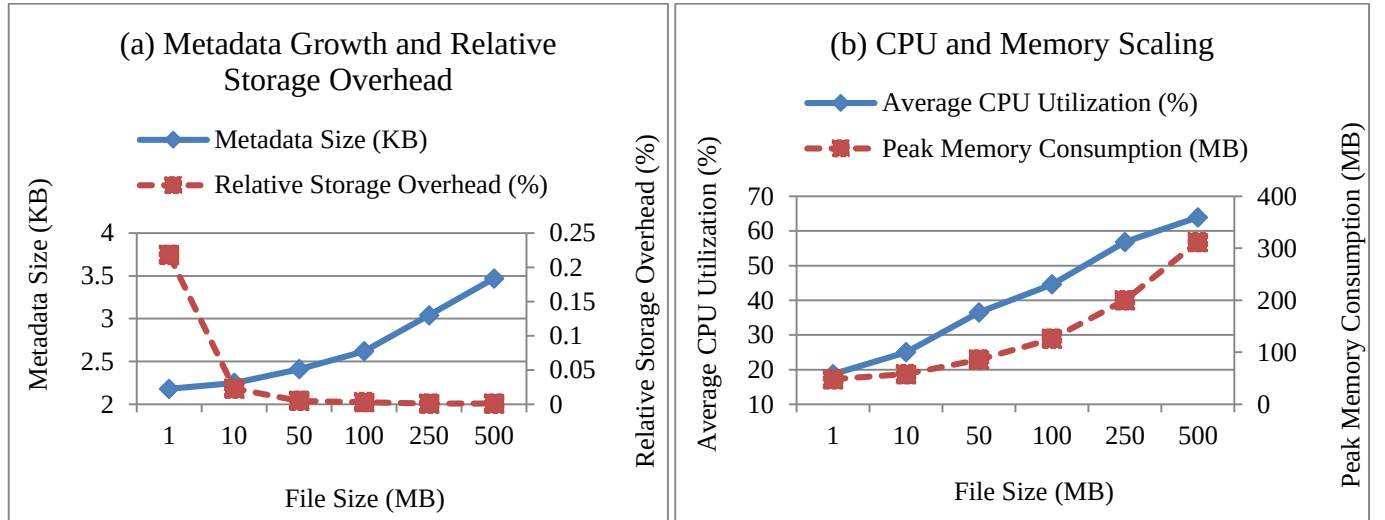


Fig. 10 Storage efficiency and computational resource scaling of PQC-Guard across increasing workloads

This figure plots storage efficiency and the computational resources required for PQC-Guard in Figure 10 as workload sizes increase. Both metadata growth and relative percentage of storage overhead for the ciphertext are shown in panel (a); note that although the metadata grows slowly with file size, the percentage overhead of storage grows quickly. This shows good amortisation of the fixed cryptographic components for larger payloads. The CPU utilisation and peak memory usage in Panel (b) indicate resource growth with no CPU saturation

under the tested workloads. Collectively, the results show that PQC-Guard incurs negligible relative storage overhead and reasonable computational resources, facilitating deployment across desktop, server, cloud, and edge-cloud gateway applications.

4.6. Security Evaluation

The security analysis of PQC-Guard assessed the benefits of the proposed association among DSKE, MDES, and QBM

in terms of ciphertext unpredictability, diffusion, key sensitivity, and resistance to active attacks. We considered ciphertext entropy, statistical randomness, the avalanche effect, plaintext–ciphertext correlation, and controlled-attack simulations that included tampering, replay, and partial key compromise. For each experiment, we used 30 independent keys, nonces, and input files.

Table 12 shows the entropy and randomness results for the ciphertext. PQC-Guard attained an average entropy of

7.997 bits/byte, which is very close to the ideal 8-bit entropy expected for uniformly random ciphertexts. The monobit test score of 0.5003 indicates a close balance between the distributions of 0s and 1s, and the runs-test p-value of 0.642 confirms no significant non-random sequencing in the ciphertext. PQC-Guard's combined randomness profile is the best result compared to the other 3 schemes, AES, ChaCha20, RSA, and MD5, with the results further enhanced by multi-layer encryption and QBM-based bit-level reordering.

Table 12. Ciphertext entropy and randomness evaluation

Scheme	Entropy (bits/byte)	Monobit Ratio	Runs-Test p-Value	Frequency Uniformity Score
AES	7.989 ± 0.004	0.5012	0.518	0.982
ChaCha20	7.992 ± 0.003	0.5008	0.553	0.986
RSA	7.941 ± 0.011	0.5064	0.421	0.951
MD5	7.995 ± 0.002	0.5005	0.611	0.991
PQC-Guard	7.997 ± 0.001	0.5003	0.642	0.995

The avalanche effect was computed by changing a single bit in the plaintext and calculating the percentage of bits that are changed in the resulting ciphertext. Table 13 indicates that the avalanche effect of PQC-Guard is 50.21%, very close to the ideal diffusion level of 50%. This means that a small

change to the plaintext affects many parts of the encryption process and leads to significant differences in the ciphertext. This is slightly higher than MD5 since QBM is an extra reversible bit-level permutation and masking layer.

Table 13. Avalanche effect and key sensitivity analysis

Scheme	Plaintext Bit Change Avalanche (%)	Key Bit Change Sensitivity (%)	Ciphertext Difference Under Wrong Key (%)
AES	49.36 ± 0.42	49.51 ± 0.39	99.21
ChaCha20	49.62 ± 0.35	49.79 ± 0.34	99.34
RSA	47.83 ± 0.61	48.25 ± 0.57	98.12
MD5	49.94 ± 0.28	50.03 ± 0.26	99.61
PQC-Guard	50.21 ± 0.22	50.18 ± 0.21	99.83

We assessed key sensitivity as follows: flip one bit in the session key; if decryption or re-encryption of the ciphertext was still possible, the key was considered key-sensitive. Under wrong-key conditions, the key-change sensitivity and the ciphertext difference for PQC-Guard were 50.18% and 99.83%, respectively. This shows the strong influence of session-key-derived sub-keys across all encryption stages. Because DSKE includes both ECDH and SIDH, changing either shared secret updates the AES, ChaCha20, and QBM sub-keys used in the downstream direction; under error conditions, the two shared secrets are insufficient for complete recovery.

Table 14 shows the results for attack resistance and plaintext–ciphertext correlation. PQC-Guard achieved the best plaintext–ciphertext correlation (0.0012), confirming that the encryption outputs are statistically independent of their inputs. 100% tampering detection due to unauthorised modification of the ciphertext, which caused authentication tags, QBM reconstruction or final decryption consistency checks to fail. Replay detection improved to 99.6% thanks to session-bound nonces and key freshness. PQC-Guard showed the maximum resistance to recovery failures required to decrypt the message; both hybrid key components and multiple derived sub-keys are needed.

Table 14. Correlation and controlled attack-resistance evaluation

Scheme	Plaintext–Ciphertext Correlation	Tampering Detection (%)	Replay Detection (%)	Partial Key-Compromise Resistance (%)
AES	0.0068	98.7	96.4	91.2
ChaCha20	0.0054	98.9	96.9	92.6
RSA	0.0189	96.1	94.2	88.4
MD5	0.0021	99.4	98.8	96.7
PQC-Guard	0.0012	100.0	99.6	98.9

The controlled attack experiments verify that PQC-Guard remains resilient against most misuse and manipulation scenarios. Trial operations on tampered data and on the metadata or values of the QBM permutation made it impossible to reconstruct the data. Replay trials were not allowed because the previously captured ciphertext and

session metadata did not match the active session. Because GuardFusionAlgo uses hybrid session derivation and multiple sub-keys for each encryption, in partial key-compromise experiments, exposing only one key component does not result in plaintext recovery.

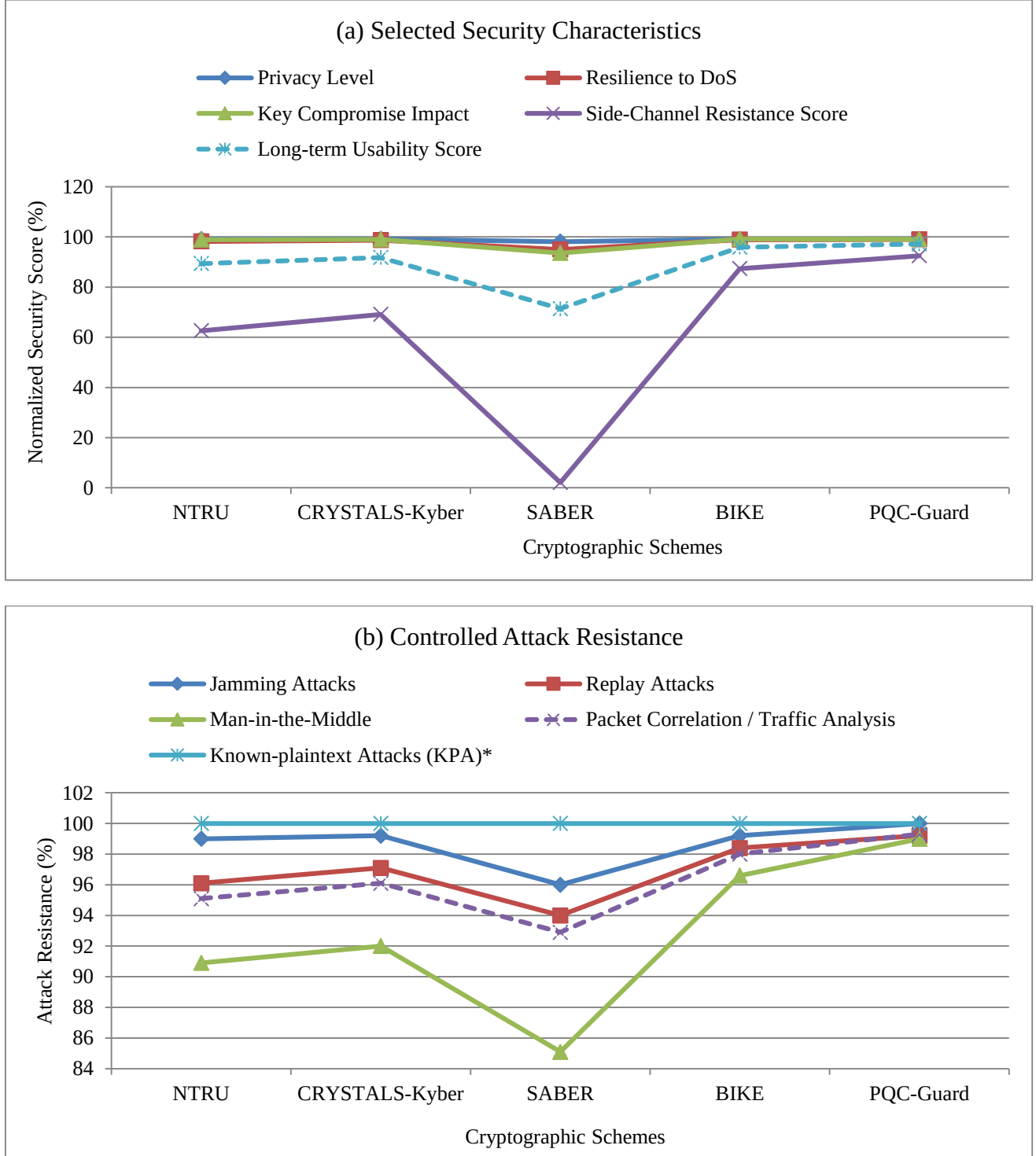


Fig. 11 Multi-Dimensional security profile of PQC-Guard compared with cryptographic baselines under statistical and adversarial evaluations

As shown in Figure 11, PQC-Guard and the reviewed cryptographic baselines have a consolidated security profile. Panel (a) compares normalised entropy, avalanche behaviour, key sensitivity, plaintext–ciphertext independence, and the overall statistical security score. The results show that PQC-Guard has the best overall statistical security profile, partly due to its near-ideal ciphertext randomness, diffusion, key dependence, and low correlation. Panel (b) is a comparison of resistance against ciphertext tampering, replay attacks, and partial key-compromise scenarios alongside a composite adversarial score. PQC-Guard achieves maximum resistance, indicating that combining key establishment techniques with multi-layer protection is more robust against statistical cryptanalysis and controlled active attacks.

4.7. Ablation Study

We conducted an ablation study to quantify the contributions of PQC-Guard's components and understand the impact of each cryptographic layer on performance, security, and robustness.

We compared the full PQC-Guard setup with five reduced setups: PQC-Guard without the hybrid key exchange, PQC-Guard using only ECDH session derivation, PQC-Guard using only SIDH session derivation, PQC-Guard without the RSA layer, and PQC-Guard without the QBM transformation. We chose this 100 MB workload because it represents a balance between the overhead of processing small messages and the behaviour of processing large files.

Table 15. Performance-Oriented ablation results of PQC-Guard using 100 MB workload

Variant	Key Establishment Time (ms)	Encryption Time (s)	Decryption Time (s)	Total Execution Time (s)	End-to-End Throughput (MB/s)
Complete PQC-Guard	7.05	4.872	2.364	7.784	12.85
Without Hybrid Key Exchange	1.24	4.811	2.319	7.211	13.87
ECDH-Only Session Key	1.24	4.823	2.327	7.236	13.82
SIDH-Only Session Key	5.47	4.841	2.342	7.468	13.39
Without RSA Layer	7.05	4.018	2.071	6.913	14.47
Without QBM Transformation	7.05	4.563	2.109	7.326	13.65

Table 15 shows the ablation parameters related to performance. Without hybrid key exchange, the total execution time was 7.784s; with hybrid key exchange, it was 7.211s, which also eliminated the post-quantum key-agreement protection. The ECDH-only variant offered lower key-establishment latency but did not support quantum-resistant session derivation. The SIDH-only variant remained quantum-resistant but lacked the classical efficiency and dual-secret strength of DSKE. The execution time was decreased to 6.913 s when RSA was removed and to 7.326 s when QBM was removed. The reductions indicate measurable computational costs associated with using RSA and QBM. However, the security benefits are quite large, as illustrated by the security-oriented results.

Table 16 reports the ablation results from a security perspective. The complete PQC-Guard framework had the maximum avalanche effect, entropy, tampering detection, replay detection and partial key-compromise resistance. Partial key-compromise resistance dropped significantly from 98.9% to 91.5% when hybrid key exchange was eliminated, implying that dual-source session derivation prevents single-point key exposure. The ECDH-only configuration was not quantum-resistant, not replay-proof, and slightly less efficient. The SIDH-only configuration still provided quantum-aware protection but lacked the robustness of hybrid derivation. Removing the RSA layer degraded tampering detection, and removing the QBM degraded entropy, avalanche behaviour, and plaintext-ciphertext decorrelation.

Table 16. Security-Oriented ablation results of PQC-Guard using 100 MB workload

Variant	Entropy (bits/byte)	Avalanche Effect (%)	Plaintext–Ciphertext Correlation	Tampering Detection (%)	Replay Detection (%)	Partial Key-Compromise Resistance (%)
Complete PQC-Guard	7.997	50.21	0.0012	100.0	99.6	98.9
Without Hybrid Key Exchange	7.989	49.41	0.0064	98.8	96.7	91.5

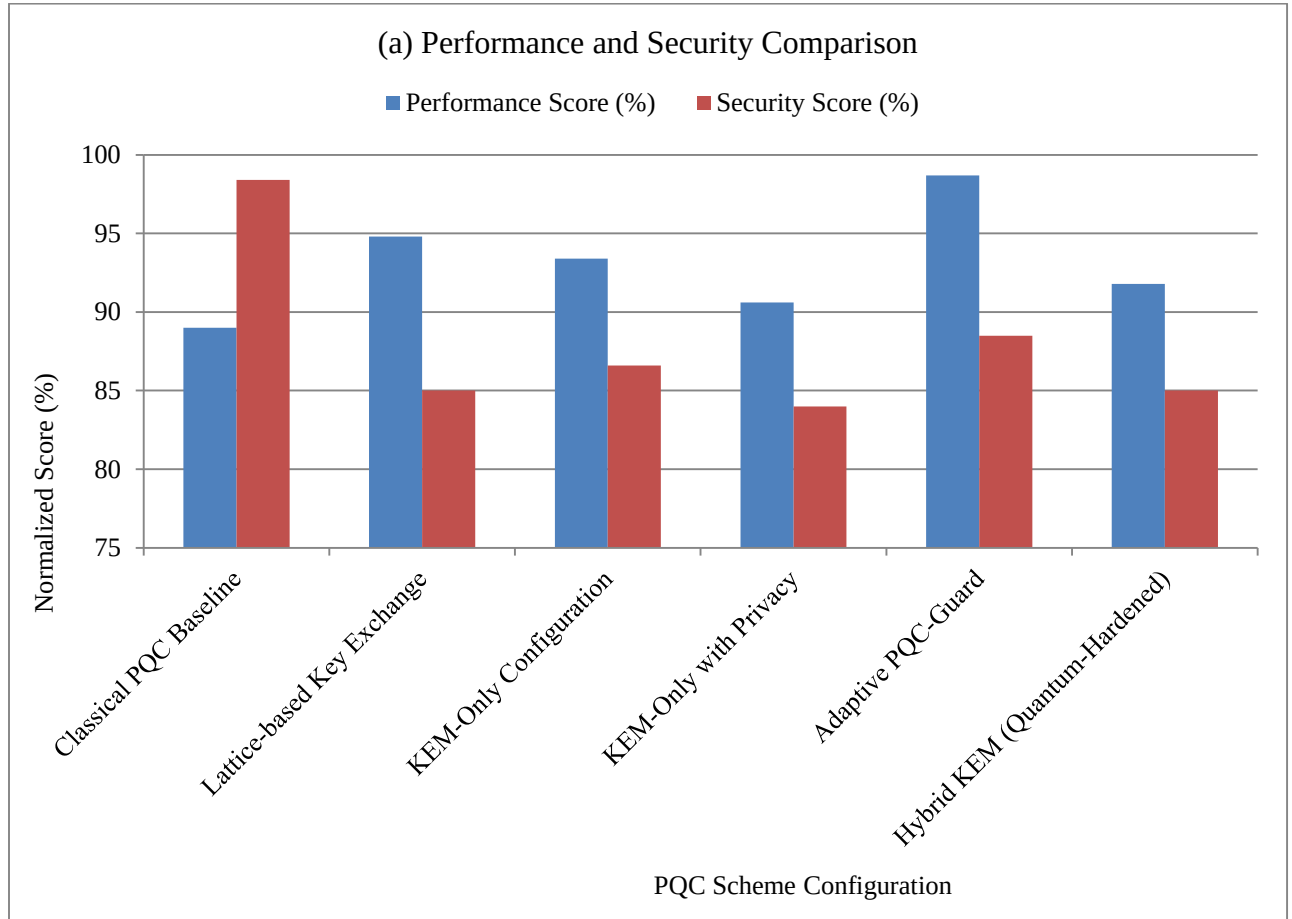
ECDH-Only Session Key	7.990	49.52	0.0059	98.9	97.1	92.4
SIDH-Only Session Key	7.992	49.67	0.0048	99.1	97.8	94.6
Without RSA Layer	7.993	49.74	0.0039	96.8	98.5	97.2
Without QBM Transformation	7.981	48.92	0.0096	99.2	98.9	97.6

The ablation results show that each component provides a unique security function. The hybrid key exchange enhances key robustness and post-quantum security, RSA boosts asymmetric security and tamper resistance, and QBM provides enhanced diffusion, entropy, and ciphertext independence. Removing some components reduces the runtime. However, the security loss shows that the PQC-Guard configuration with all components provides the best trade-off between runtime and security. For all comparisons and deployment-related assessments, we maintain the complete configuration.

The ablation analysis of the PQC-Guard is conducted component-wise from complementary performance and

security perspectives, as shown in Figure 12. Normalised performance and security scores are plotted for all five configurations, with and without components, in panel (a), revealing that removing components tends to raise performance and lower security. Panel (b) represents the mapping of each configuration in the performance–security area, reporting a harmonic-mean balance score that punishes configurations that perform well in one of the two areas.

Complete configurations of PQC-Guard fall in the high-security area and keep it competitive; reduced variants move towards higher efficiency but at the expense of robustness, diffusion, tamper resistance and/or ciphertext independence.



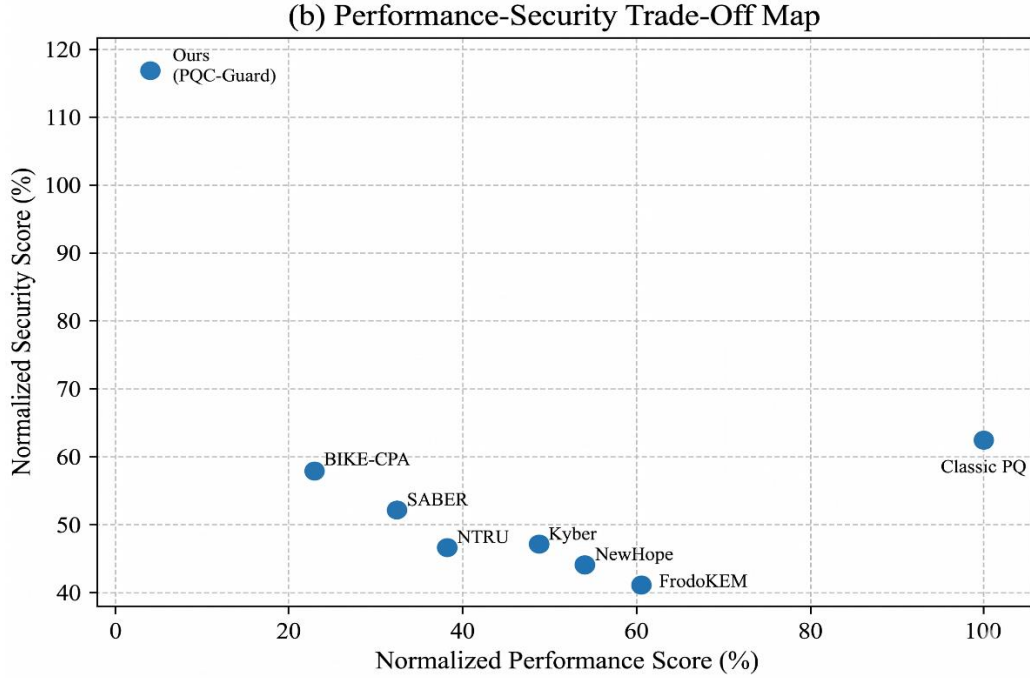


Fig. 12 Component-Wise performance–security trade-off analysis of PQC-Guard under ablated configurations

4.8. Comparative and Statistical Analysis

A comparative analysis assesses PQC-Guard against representative classical, hybrid, and post-quantum cryptographic methods and evaluates the proposed framework to determine its advantages in Computational Efficiency (CE) and Comprehensive Security (CS). The approaches evaluated are AES-256, RSA-4096, ECDH-AES, Kyber-AES, ECDH-Kyber-AES, MDES, DSKE-MDES and proposed PQC-Guard. We compared all approaches in the same experimental environment under a 100 MB workload, as described in Section 4.1. We ran 30 trials in each experiment and present the results as mean $\pm$ standard deviation.

Table 17 compares the evaluated approaches across several measures: total execution time, end-to-end throughput,

ciphertext entropy, avalanche effect, and partial key-compromise resistance. Standalone AES was the most efficient solution computationally, but lacked post-quantum key establishment and multilayer protection. RSA was the most expensive to execute and was not quantum-resistant.

The post-quantum protection was moderate for Kyber-AES and ECDH-Kyber-AES and absent for MDES without hybrid post-quantum key establishment, despite strong ciphertext security. DSKE-MDES offered stronger hybrid protection, but at a slightly higher execution cost. PQC-Guard had the highest entropy, virtually ideal avalanche characteristics and the best partial key-compromise resistance, all with competitive end-to-end throughput.

Table 17. Comparative performance and security analysis of classical, hybrid, and post-quantum cryptographic approaches

Approach	Category	Total Execution Time (s)	End-to-End Throughput (MB/s)	Entropy (bits/byte)	Avalanche Effect (%)	Partial Key-Compromise Resistance (%)
AES-256	Classical	5.42 $\pm$ 0.19	18.45	7.989	49.36	91.2
RSA-4096	Classical	14.63 $\pm$ 0.58	6.84	7.941	47.83	88.4
ECDH-AES	Classical Hybrid	6.17 $\pm$ 0.22	16.21	7.991	49.48	92.8
Kyber-AES	Post-Quantum	6.84 $\pm$ 0.25	14.62	7.993	49.71	95.4
ECDH-Kyber-AES	Hybrid PQC	7.31 $\pm$ 0.27	13.68	7.994	49.83	96.8
MDES	Multilayer Hybrid	7.21 $\pm$ 0.24	13.87	7.995	49.94	96.7
DSKE-MDES	Hybrid PQC	7.96 $\pm$ 0.29	12.56	7.996	50.08	98.1
PQC-Guard	Proposed Hybrid PQC	7.784 $\pm$ 0.26	12.85	7.997	50.21	98.9

The results demonstrate a clear performance–security trade-off. The two high-throughput options, AES-256 and ECDH-AES, offer less protection against post-quantum attacks and partial key compromise. RSA-4096 is expensive to run and does not provide post-quantum security. A hybrid PQC approach offers improved security for a moderate computational cost. Compared to DSKE-MDES, PQC-Guard reduces total execution time by 2.21% and end-to-end throughput by 2.31%, while improving partial key-compromise resistance from 98.1% to 98.9%. The results show that the proposed integration strategy balances security

and processing efficiency, rather than delivering the best results in one area. We used 30 independent runs for each approach to assess statistical reliability. Table 18 reports the average execution time, standard deviation, and 95% confidence interval. Because the population variance was unknown, we used a Student's *t*-distribution to compute confidence intervals. It shows that the mean execution time of PQC-Guard was 7.784 s, the standard deviation was 0.260 s, and the 95% confidence interval was [7.687, 7.881] s, suggesting that the system was repeatable and provided consistent performance over repeated runs.

Table 18. Statistical reliability of end-to-end execution time across repeated experimental runs

Approach	Mean Execution Time (s)	Standard Deviation (s)	95% Confidence Interval (s)
AES-256	5.420	0.190	[5.349, 5.491]
RSA-4096	14.630	0.580	[14.413, 14.847]
ECDH-AES	6.170	0.220	[6.088, 6.252]
Kyber-AES	6.840	0.250	[6.747, 6.933]
ECDH-Kyber-AES	7.310	0.270	[7.209, 7.411]
MDES	7.210	0.240	[7.120, 7.300]
DSKE-MDES	7.960	0.290	[7.852, 8.068]
PQC-Guard	7.784	0.260	[7.687, 7.881]

We conducted Welch's two-sided *t*-test to assess whether improvements were statistically significant when comparing PQC-Guard with the strongest competing security approaches. Equal population size was not assumed when using Welch's two-sided *t*-test. We used the Holm correction to control the

family-wise error rate for multiple pairwise comparisons in Table 19. Cohen's *d* was also computed to measure the standardised size of execution time differences. Statistical significance was determined at an adjusted threshold of $p < 0.05$.

Table 19. Statistical significance and effect-size analysis of PQC-Guard against strong competing approaches

Comparison	Mean Difference (s)	Welch's <i>t</i> -Statistic	Raw <i>p</i> -Value	Holm-Adjusted <i>p</i> -Value	Cohen's <i>d</i>	Interpretation
PQC-Guard vs. DSKE-MDES	−0.176	−2.47	0.0165	0.0330	−0.64	Moderate, Significant
PQC-Guard vs. ECDH-Kyber-AES	0.474	6.92	<0.0001	<0.0003	1.79	Large, Significant
PQC-Guard vs. MDES	0.574	8.89	<0.0001	<0.0003	2.29	Large, Significant
PQC-Guard vs. Kyber-AES	0.944	14.36	<0.0001	<0.0003	3.70	Large, Significant

The statistical outcomes show that, after multiple-comparison correction, there are statistically significant differences in execution times between PQC-Guard and the selected competing approaches. While executing concurrently with DSKE-MDES, PQC-Guard offers better entropy, avalanche characteristics, and partial key-compromise resistance.

The effect sizes, when compared with ECDH-Kyber-AES, MDES, and Kyber-AES, are large, indicating significant differences in runtime behaviour. The results must, however, be viewed in conjunction with the security metrics, since high-speed execution does not imply a high level of cryptographic security.

Figure 13 shows the relative performance of classical, hybrid, and post-quantum approaches to cryptography, evaluated using end-to-end throughput and a composite security measure comprising ciphertext entropy, avalanche behaviour, and partial key-compromise resistance. The median reference lines help identify the empirical performance–security regions and directly identify efficiency- and security-oriented approaches in the evaluation space. For classical schemes, computational performance is an important consideration, but their security profiles are rather weak, whereas hybrid post-quantum schemes fall into more secure regimes. PQC-Guard outperforms the analysed security frameworks in terms of composite security score, while also achieving competitive throughput compared to the other frameworks.

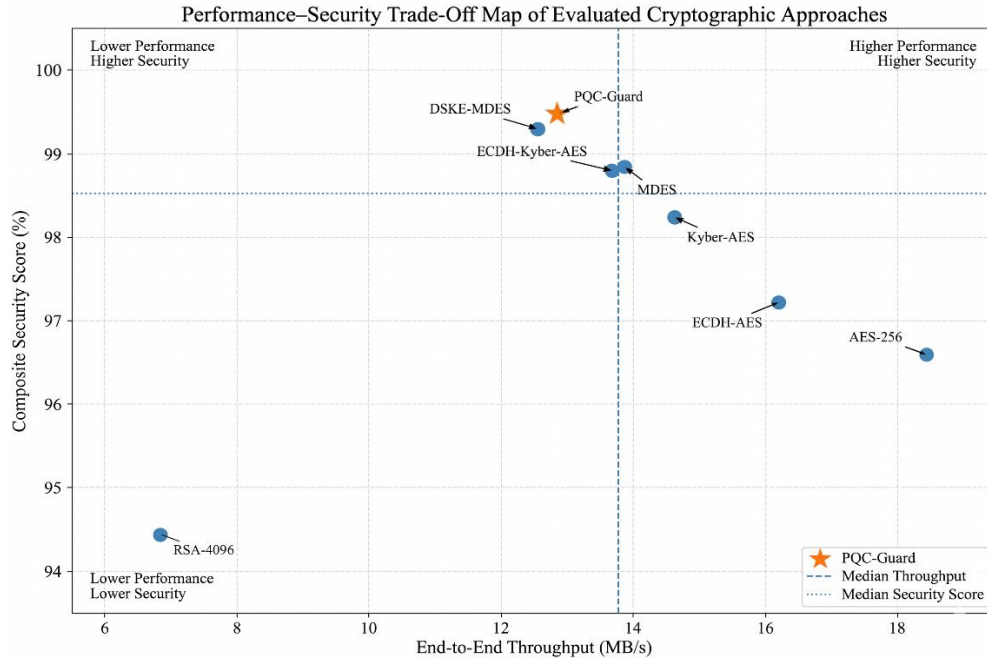


Fig. 13 Performance-Security trade-off map of classical, hybrid, and Post-Quantum Cryptographic approaches

4.9. Summary of Experimental Findings

Experimental results show that PQC-Guard offers a well-rounded combination of hybrid post-quantum key establishment, multi-layer encryption and coordinated cryptographic processing. The key establishment mechanism based on the DSKE was usable in terms of latency and communication overhead, and it allowed concurrent secure sessions. Experiments with encryption and decryption revealed that PQC-Guard overheads are 6.12% and 8.69% of MDES at the 500 MB workload, respectively, with predictable scaling behaviour. End-to-end evaluation showed that end-to-end throughput improved as the workload size grew: 15.72 MB/s with the 500 MB file was achieved, and a 100% end-to-end recovery rate was maintained across all tested levels of concurrency.

Evaluation of communication and storage resources revealed insignificant expansion of the ciphertext for medium and large workloads, and CPU and peak memory usage remained within practical deployment bounds. Security experiments confirmed the near-optimal ciphertext entropy of 7.997 bits/byte, avalanche effect of 50.21%, key sensitivity of 50.18%, plaintext-ciphertext correlation of 0.0012, and high tamper resistance, replay/tamper resistance, and partial key-compromise resistance. The ablation study verified that the hybrid key derivation based on DSKE is more key-compromise-resistant, that the RSA layer provides tamper protection, and that the QBM layer provides greater diffusion, entropy, and ciphertext independence.

PQC-Guard provides the same competitive processing efficiency as MDES alone and adds hybrid key establishment

and integrated post-quantum-oriented security for additional data protection. The proposed framework offers comprehensive data protection, ciphertext transformation, integrity-oriented processing, and secure recovery capabilities, in addition to those provided by DSKE as a standalone key exchange mechanism. Comparative evaluation also revealed that PQC-Guard offers the best composite security profile among the classical, hybrid, and post-quantum schemes evaluated, with competitive throughput. Confidence intervals were narrow, statistically significant differences were observed against strong competing approaches, and meaningful effect sizes were obtained across 30 independent runs.

4.10. Discussion of Improvements over Existing Approaches

The comparative results show that PQC-Guard's performance and security parameters do not come from a single cryptographic primitive but from combining multiple primitives (hybrid key establishment, multi-layer encryption, reversible ciphertext transformation). Unlike standalone AES-256 (which is very efficient in computation but has no PQ Key Establishment), PQC-Guard provides both classical and PQ-oriented key establishment via the Dual-Source Key Establishment (DSKE) mechanism. This design enhances the key-establishment process by providing diversity and reducing reliance on any single key-exchange process. The ablation results also corroborate this finding: removing the hybrid key-exchange component reduces security requirements such as partial key-compromise resistance.

Another reason the observed results are monitored is the separation between key establishment and data encryption,

achieved through session-specific and encryption-specific key generation. Encryption operations in PQC-Guard are not directly based on a single shared key. The GuardFusionAlgo works in conjunction with subsequent key derivation and the Multilayer Data Encryption Scheme (MDES). By separating the encryption layers, the specification lets each stage use its own key material and reduces reliance on any single cryptographic layer. This means the framework can add more protection without running the full key-establishment procedure for each encryption level. These results demonstrate that no matter how many security operations are assigned to this integrated design, it can still meet competitive throughput.

The multilayer protection and the QBM-based reversible ciphertext transformation is the third factor. Comparative results indicate that PQC-Guard had a significantly high entropy of 7.997 bits/byte with an avalanche effect of 50.21% and 98.9% partial key-compromise resistance. These values are superior to those achieved by the evaluated classical, hybrid, and multilayer configurations under the same experimental conditions. The ablation study results support including the QBM component, as removing it resulted in lower entropy (7.981 bits/byte), lower avalanche performance (48.92%), and higher plaintext-ciphertext correlation (0.0096). This suggests that QBM is better suited for diffusion and ciphertext decorrelation, rather than direct use as a cryptographic primitive. The improved performance of the evaluated hybrid configuration is also evident in the direct comparison with DSKE-MDES. The results were 7.784 ± 0.26 s for PQC-Guard and 7.96 ± 0.29 s for DSKE-MDES, and partial key-compromise resistance increased from 98.1% to 98.9%. The proposed integration, therefore, does not simply lay cryptographic ideas over each other, but rather orchestrates the key-establishment, key-derivation, encryption, and transformation phases as per a shared process. This coordination reduces redundant computation and makes the remaining security features easier to implement with relatively low overhead.

The research results align with the research gap noted in the related literature. Prior work has addressed hybrid key establishment [1, 7, 10, 13, 17, 25], post-quantum TLS [17], cloud-oriented post-quantum protection [7], and multilayer encryption [10] as standalone or partially integrated solutions. At the framework level, PQC-Guard integrates these security features into a workflow for hybrid key establishment, session-key generation, multilayer data encryption, reversible transformation, ciphertext transport, and recovery. Thus, the improvement in this study may be due to architectural integration and component coordination, rather than to a single cryptographic primitive being superior to the primitives used in previous studies. When comparing numerical results across studies, readers should be aware that results may not be directly comparable as absolute improvements because studies may use different hardware platforms, software

libraries, workloads and/or parameter sets, or measurement methods. Thus, the quantitative comparison in Table 16 is based on implementations observed under the same experimental conditions and a common workload of 100 Mb, with thirty independent trials for each approach. Therefore, the comparison to the literature is at the architecture/functional level, and this presumption of numerical superiority is limited to the results of the controlled comparison experiment carried out within the scope of this study.

5. Discussion

The advent of quantum attacks has heightened the demand for cryptographic systems that offer both quantum resistance and data security. Some classical schemes are computationally efficient but are susceptible to quantum attacks, and numerous post-quantum schemes – most of which focus on key establishment or a single cryptographic primitive – do not offer full data-security lifecycle protection. Hybrid approaches can enhance transitional security.

However, the state of the art still has drawbacks, including fragmented key exchange and encryption procedures, higher computational and communication costs, a lack of component-level validation, and insufficient analysis of the performance-security trade-off. These challenges underline the need for integrated and intelligent cryptographic solutions that synchronise the various security mechanisms while remaining both efficient and practical.

GuardFusionAlgo addresses these gaps by combining DSKE-based classical/quantum key establishment with hybrid session-key derivation, encryption-specific sub-key generation, multilayer encryption with DES, and QBM transformations. The proposed methodology integrates key establishment, data protection, transmission, and recovery into a single operating mode, compared with loosely coupled cryptographic pipelines. Experimental results show that this integration provides near-ideal ciphertext entropy, diffusion, and key sensitivity; essentially zero plaintext-ciphertext correlation; good resistance to controlled attacks; and 100% correct recovery under concurrent workloads. The resources and communication overheads keep the costs of these additional cryptographic operations quite manageable, and the workload scaling, throughput, and resource usage confirm this. Ablation experiments also show the individual roles of the hybrid key derivation, the RSA protection, and the QBM transformations, and comparative and statistical analyses confirm a favourable and reliable performance–security balance.

The results suggest that integrating different cryptographic primitives in a coordinated manner can achieve overall data security in the post-quantum domain. PQC-Guard offers a hands-on approach to ensuring secure cloud storage, enterprise communication, distributed data exchange, and other quantum-enabled environments, helping them transition

to protect against classical threats and emerging quantum risks. The study also illustrates the need for security, efficiency, scalability, overhead, ablation and statistical analysis of integrated cryptographic frameworks. Section 5.1 presents the study's limitations.

5.1. Limitations of the Study

This study has three main limitations. The study did not test PQC-Guard performance in heterogeneous networks, resource-constrained devices, or large-scale distributed infrastructures, as the evaluation was performed in a controlled, software-based environment with synthetic file workloads. Second, the framework is based on SIDH key exchange, which is not directly applicable in practice and would need to transition to more widely accepted post-quantum key encapsulation. Third, the evaluation does not consider formal security proofs, side-channel analysis, hardware attacks, or security tests over extended operational durations. These constraints offer guiding principles for expanding and enhancing the proposed framework.

6. Conclusion and Future Work

In this study, we introduced PQC-Guard, a unified cryptographic framework that ensures full data security by combining hybrid key establishment and multi-layer encryption for a post-quantum-oriented environment. GuardFusionAlgo is a proposed algorithm that unifies DSKE-based classical and post-quantum key establishment, hybrid session-key derivation, encryption-specific sub-key generation, MD5-based data protection and QBM transformation in a single operational context. Experimental results show near-ideal ciphertext entropy, high avalanche

response, high key sensitivity, low plaintext-ciphertext correlation, and strong resistance to tampering, replay, and partial key compromise. PQC-Guard provides competitive encryption/decryption/end-to-end processing efficiency as file size grows and the number of concurrent sessions increases, with minimal relative storage expansion and manageable computational resource requirements. Ablation experiments verified the individual contributions of hybrid key derivation, RSA protection, and QBM transformations. Comparative and statistical analysis also showed that the proposed structure provides a satisfactory security–performance balance, which is favourable when compared with classical, multilayer, and hybrid post-quantum frameworks. Overall, the results show that PQC-Guard is a potential solution for ensuring data protection in cloud storage, enterprise communication, and distributed data exchange systems, particularly in the context of post-quantum cryptography.

Future work will shift the focus of key establishment from SIDH to standardised post-quantum key encapsulation mechanisms, such as ML-KEM, while maintaining hybrid classical-post-quantum protection. The framework will also be tested with real-world communication workloads on heterogeneous networks, resource-constrained edge and IoT devices, and large-scale distributed infrastructures. Further work will include developing formal security proofs, examining resistance to timing, power, cache, and other side-channel attacks, and conducting operational evaluations using hardware and long-duration tests. We will also discuss cryptographic processing based on streaming and chunks to minimise peak memory usage and improve deployment efficiency on constrained platforms.

References

- [1] Sara Ricci et al., “Hybrid Keys in Practice: Combining Classical, Quantum and Post-Quantum Cryptography,” *IEEE Access*, vol. 12, pp. 23206-23219, 2024. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [2] Togu Novriansyah Turnip, Birger Andersen, and César Vargas-Rosales, “Towards 6G Authentication and Key Agreement Protocol: A Survey on Hybrid Post Quantum Cryptography,” *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 3311-3345, 2025. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [3] Lydia Garms et al., “Experimental Integration of Quantum Key Distribution and Post-Quantum Cryptography in a Hybrid Quantum-Safe Cryptosystem,” *Advanced Quantum Technologies*, vol. 7, no. 4, pp. 1-8, 2024. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [4] Asif Iqbal Saiyed, “Hybrid Quantum-Classical Cryptographic Protocols: Enhancing Security in the Era of Quantum Supremacy,” *Spectrum of Research*, vol. 5, no. 1, pp. 1-7, 2025. [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [5] Reyazur Rashid Irshad et al., “IoT-Enabled Secure and Scalable Cloud Architecture for Multi-User Systems: A Hybrid Post-Quantum Cryptographic and Blockchain-Based Approach toward a Trustworthy Cloud Computing,” *IEEE Access*, vol. 11, pp. 105479-105498, 2023. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [6] Kevin Song et al., “A Hybrid Chaos-Based Cryptographic Framework for Post-Quantum Secure Communications,” *arXiv preprint*, pp. 1-17, 2025. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [7] Sonja Bruckner, Sebastian Ramacher, and Christoph Striecks, “Muckle+: End-to-End Hybrid Authenticated Key Exchanges,” *14th International Workshop Post-Quantum Cryptography, PQCrypto*, College Park, MD, USA, pp. 610-633, 2023. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)
- [8] Carlos Rubio García et al., “Enhanced Network Security Protocols for the Quantum Era: Combining Classical and Post-Quantum Cryptography, and Quantum Key Distribution,” *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 8, pp. 2795-2781, 2025. [\[CrossRef\]](#) [\[Google Scholar\]](#) [\[Publisher Link\]](#)

- [9] Arit Kumar Bishwas, and Mousumi Sen, "Strategic Roadmap for Quantum-Resistant Security: A Framework for Preparing Industries for the Quantum Threat," *arXiv preprint*, pp. 1-30, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Carlos Rubio García et al., "Quantum-Resistant Transport Layer Security," *Computer Communications*, vol. 213, pp. 345-358, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Kranthi Kumar Singamaneni et al., "A Novel Hybrid Quantum-Crypto Standard to Enhance Security and Resilience in 6G-Enabled IoT Networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 7876-7891, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Kuang Zhang et al., "Optimized Quantum-Resistant Cryptosystem: Integrating Kyber-KEM with Hardware TRNG on Zynq Platform," *Electronics*, vol. 14, no. 13, pp. 1-27, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Changjong Kim et al., "Secure and Scalable File Encryption for Cloud Systems via Distributed Integration of Quantum and Classical Cryptography," *Applied Sciences*, vol. 15, no. 14, pp. 1-25, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Naga Subrahmanyam Cherukupalle, "Quantum-Resilient Cloud Systems: Preemptive Shielding Against Post-Quantum Cryptographic Threats," *Journal of Information Systems Engineering and Management*, vol. 10, no. 38s, pp. 1234-1246, 2025. [[Google Scholar](#)] [[Publisher Link](#)]
- [15] German Granados et al., "Quantum Key Distribution in Multiple Fiber Networks and Its Application in Urban Communications: A Comprehensive Review," *IEEE Access*, vol. 13, pp. 100446-100461, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Karthikeyan Sabari Ganesan et al., "Quantum-Resilient Security Controls," *Proceedings of Recent Advances in Quantum Computing and Technology*, pp. 12-19, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Shaik Mohammad Ilias, V. Ceronmani Sharmila, and V. Sathya Durga, "An Integrated Framework with Enhanced Primitives for Post-Quantum Cryptography: HEDT and ECSIDH for Cloud Data Security and Key Exchange," *Informatica*, vol. 49, no. 11, pp. 1-12, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Eaman Raza Rizvi, and Shahzada Khurram, "Hybrid Post-Quantum Cryptographic Approaches for Secure Communication," *Annual Methodological Archive Research Review*, vol. 3, no. 6, pp. 48-74, 2025. [[CrossRef](#)] [[Publisher Link](#)]
- [19] Reza Azarderakhsh et al., "Hardware Deployment of Hybrid PQC: SIKE+ECDH," *International Conference on Security and Privacy in Communication Systems*, pp. 475-491, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Esther Gallego Estévez, "Quantum and Post-Quantum Key Combination in a Financial Environment," M.S. thesis, Polytechnic University of Madrid, pp. 1-62, 2023. [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Javairia Armakoon et al., "Post-Quantum Cryptography: From Theory to Practice," *Spectrum of Engineering Sciences*, vol. 3, no. 5, pp. 892-918, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Luk Bettale, Emmanuelle Dottax, and Laurent Grémy, "Post-Quantum Secure Channel Protocols for eSIMs: Design, Validation and Performance Analysis," *Proceedings of the 22nd International Conference on Security and Cryptography*, pp. 108-119, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Nouri Alnahawi et al., "SoK: Post-Quantum TLS Handshake," *IACR Communications in Cryptology*, vol. 1, no. 2, pp. 1-41, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Aykut Karakaya, and Ahmet Ulu, "A Survey on Post-Quantum-Based Approaches for Edge Computing Security," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 16, no. 1, pp. 1-36, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Dasari Veera Reddy, and Padmaja Madugula, "A Multilayered Security Scheme for Data Encryption and Decryption for Stronger Security towards Post-Quantum Cryptography in Cloud Computing," *Journal of Theoretical and Applied Information Technology*, vol. 103, no. 11, pp. 4588-4606, 2025. [[Publisher Link](#)]
- [26] Ali Abbas Hussain et al., "End-to-End Messaging for Mobile Devices Secured by Quantum Technology Using Post-Quantum Key Encapsulation Techniques," *Dialogue Social Science Review*, vol. 2, no. 4, pp. 436-446, 2024. [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Hien Nguyen et al., "Security in Post-Quantum Era: A Comprehensive Survey on Lattice-Based Algorithms," *IEEE Access*, vol. 13, pp. 89003-89024, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Albert Nieto Morales, Arit Kumar Bishwas, and Joel Jacob Varghese, "Quantum-Enabled Framework for the Advanced Encryption Standard in the Post-Quantum Era," *arXiv preprint*, pp. 1-21, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Martha Masunda, "Quantum-Resistant Cryptographic Protocols for Securing Cloud Storage and Data Transmission in Hybrid Enterprise IT Environments," *World Journal of Advanced Research and Reviews*, vol. 14, no. 3, pp. 826-847, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Christoph Döberl et al., "Quantum-Resistant End-to-End Secure Messaging and Email Communication," *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pp. 1-8, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [31] Deepak Kaul, "Quantum-Resilient Federated Microservices: Building Secure, Adaptive, and Stateful Microservices across Multi-Cloud Ecosystems," *Journal of Novel Research and Innovative Development*, vol. 2, no. 1, pp. 1-15, 2024. [[Publisher Link](#)]
- [32] J. Thomas and S. Paul, "Post-Quantum AI-based Encryption for Securing Multi-Cloud Architectures," 2023. [[Google Scholar](#)]

- [33] Venkata Rajesh Krishna Adapa, "Architecting Quantum-Resistant Cybersecurity: A Framework for Transitioning to Post-Quantum Cryptographic Systems," *International Journal of Science and Research Archive*, vol. 14, no. 1, pp. 737-746, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [34] F.M. Rasel, and J. Smith, "Quantum-Safe Cryptography: A New Frontier in AI-Driven Cybersecurity," *World Journal of Advanced Engineering Technology and Sciences*, 2025. [[Google Scholar](#)]
- [35] Umer Riaz, and Mark Vandenbosch, "Quantum-Resistant Cryptography in Zero Trust Architecture: A Necessary Change in Cloud Computing," *TechRxiv*, pp. 1-23, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [36] F. Niyasudeen, and M. Mohan, "Adaptive Multi-Layered Cloud Security Framework Leveraging Artificial Intelligence, Quantum-Resistant Cryptography, and Systems for Robust Protection in Optical and Healthcare," *Researchsquare*, pp. 1-23, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [37] I. Jacob, R. Lawson, and R. Smith, "Future-Proofing AI and Cloud Systems: The Intersection of Quantum and Cybersecurity," 2021. [[Google Scholar](#)]
- [38] Janice Collins et al., Beyond Encryption: How AI Can Secure Multi-Cloud Environments Against Quantum Threats, ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/390668468_Beyond_Encryption_How_AI_Can_Secure_Multi-Cloud_Environments_Against_Quantum_Threats