International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

Lifecycle-Aware Post-Quantum Cryptography with Provenance Verification for Secure Cloud Microservices

Dr. S. Nithya 1, † ,* ; Mr. S. Lohit 1, † ,*

1 Department of Information Technology, PSNA College of Engineering and Technology, Dindigul-624622, Tamil Nadu, India

† These authors contributed equally to this work.

* Corresponding Authors: snithya@psnacet.edu.in (Dr. S. Nithya); lohit.tech17@gmail.com (Mr. S. Lohit)

Author Designations: Dr. S. Nithya – Assistant Professor*; Mr. S. Lohit – B.Tech Scholar*

Abstract:

The transition from classical cryptographic schemes to post-quantum cryptography (PQC) in cloud microservices introduces significant challenges in maintaining performance, interoperability, and verifiable trust. This work presents Lifecycle-Aware Post-Quantum Cryptography with Provenance Verification for Secure Cloud Microservices (LAPQC), a novel framework that enables controlled and traceable cryptographic evolution across distributed service environments. Unlike existing approaches that rely on static replacement of cryptographic primitives, the proposed model introduces a lifecycle-aware transition mechanism combined with provenance-driven verification. The framework comprises two core modules. The Lifecycle-Aware PQC Transition Engine (LAPTE) facilitates adaptive switching between classical, hybrid, and PQC schemes based on operational context, ensuring seamless migration without service disruption. The Cryptographic Provenance and Verification Engine (CPVE) maintains verifiable records of cryptographic operations, enabling integrity validation, auditability, and protection against downgrade and substitution attacks. Experimental evaluation is conducted using metrics such as throughput and end-to-end latency under real-time workloads. Observed results indicate that while classical AES achieves maximum throughput, the proposed hybrid PQC scheme maintains competitive throughput levels while significantly improving quantum resilience. Furthermore, the hybrid approach demonstrates a balanced trade-off between performance and security compared to standalone lattice-based and hash-based PQC methods, which exhibit higher latency overheads. The integration of lifecycle-aware transition with cryptographic provenance introduces a new paradigm for secure and verifiable PQC adoption, making LAPQC a scalable and robust solution for next-generation cloud microservice architectures.

Keywords: Post-Quantum Cryptography (PQC), Cloud Microservices Security, Lifecycle-Aware Cryptographic Transition, Cryptographic Provenance, Provenance Verification, Hybrid Cryptography, Lattice-Based Cryptography, Hash-Based Cryptography, Quantum-Resilient Security, Secure Service Migration, Cryptographic Agility, Distributed Systems Security

1. INTRODUCTION

The rapid proliferation of cloud-native microservices has transformed the design of modern distributed systems, enabling scalable, modular, and independently deployable services [1]. However, this architectural evolution has also expanded the attack surface, particularly in the context of cryptographic security. Existing security mechanisms in microservices predominantly rely on classical cryptographic algorithms such as RSA and AES, which are vulnerable to emerging quantum computing capabilities [2]. Algorithms like Shor's algorithm can efficiently break widely used public-key schemes [3], while Grover's algorithm reduces the effective security strength of symmetric encryption [4]. This impending threat necessitates a transition toward Post-Quantum Cryptography (PQC), which is designed to resist quantum-enabled attacks [5]. Despite significant advancements in PQC standardization, the integration of quantum-resistant algorithms into cloud microservices remains a complex challenge. Most

existing approaches adopt a static replacement strategy, where classical cryptographic primitives are directly substituted with PQC alternatives [6]. Such methods fail to address critical issues including interoperability between heterogeneous services, performance degradation due to computational overhead, and the absence of verifiable mechanisms to track cryptographic transitions [7]. In dynamic microservice environments, where services continuously evolve and interact, an unstructured migration to PQC can introduce inconsistencies, security vulnerabilities, and operational instability [8].

Another critical limitation in current systems is the lack of cryptographic transparency and traceability [9]. Microservices often operate across multiple domains and trust boundaries, making it difficult to verify the origin, integrity, and evolution of cryptographic operations. Without a formal mechanism to record and validate cryptographic lineage, systems remain susceptible to downgrade attacks, unauthorized algorithm substitution, and integrity violations [10]. This gap highlights the need for a provenance-aware security model that can ensure accountability and verifiability of cryptographic processes. To address these challenges, this work proposes Lifecycle-Aware Post-Quantum Cryptography with Provenance Verification for Secure Cloud Microservices (LAPQC). The proposed framework introduces a structured and adaptive approach for cryptographic evolution, enabling seamless transition from classical to hybrid and fully post-quantum states. The system is designed to operate within dynamic microservice environments, ensuring continuity of service while maintaining robust security guarantees.

The proposed methodology is composed of two key modules. The Lifecycle-Aware PQC Transition Engine (LAPTE) governs the controlled migration of cryptographic schemes by incorporating context-aware decision mechanisms, transition policies, and runtime adaptability. This module enables coexistence and gradual replacement of classical and PQC algorithms, ensuring minimal disruption to service operations. Complementing this, the Cryptographic Provenance and Verification Engine (CPVE) establishes a verifiable trust layer by maintaining structured records of cryptographic operations, key evolution, and algorithm transitions. This enables continuous validation of integrity, supports auditability, and mitigates risks associated with downgrade and substitution attacks. The effectiveness of the proposed framework is validated through experimental evaluation using performance metrics such as throughput and end-to-end latency under real-time workload conditions. Observed results demonstrate that while classical cryptographic schemes achieve higher raw performance, the proposed hybrid PQC approach offers a balanced trade-off between computational efficiency and quantum resilience. This balance is critical for practical deployment in cloud microservice environments where both performance and security are essential.

The main contributions of this work include the introduction of a lifecycle-aware cryptographic transition model that enables structured adoption of post-quantum cryptography in cloud microservices. A provenance-driven verification mechanism is designed to ensure traceability, integrity, and accountability of cryptographic operations across distributed environments. In addition, a hybrid PQC integration strategy is developed to achieve an effective balance between computational performance and quantum resistance. The proposed framework is further validated through comprehensive experimental evaluation, demonstrating its feasibility and effectiveness in real-time cloud microservice settings. The remainder of this paper is organized as follows. Section 2 reviews recent literature on PQC integration and microservice security. Section 3 outlines the system model and problem formulation. Section 4 presents the proposed LAPQC framework in detail. Section 5 discusses experimental results and performance analysis. Section 6 concludes the paper with future research directions.

2. EXISTING METHODS

A set of widely adopted classical and emerging post-quantum cryptographic techniques are considered as baseline methods for analysis and comparison. These methods represent different categories of security mechanisms, ranging from traditional public-key and symmetric encryption schemes to advanced quantum-resistant approaches. The selected methods include Rivest–Shamir–Adleman (RSA) [11], a classical public-key cryptographic algorithm widely used for secure key exchange; Advanced Encryption Standard (AES) [12], a symmetric encryption standard known for its efficiency and strong security in practical applications; Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) [13], which offers resistance against quantum attacks through hard lattice problems; and Post-Quantum Cryptography based on Hash Algorithms (PQCHA) [14], which leverages hash-based constructions to provide quantum-resilient security. These methods serve as reference models to evaluate

the effectiveness, performance, and security trade-offs of the proposed framework.

2.1. Rivest–Shamir–Adleman (RSA)

Rivest–Shamir–Adleman (RSA) is a foundational public-key cryptographic algorithm extensively utilized for secure communication, key exchange, and digital signature generation in distributed and cloud-based systems. It operates on the principle of asymmetric encryption, where a mathematically related key pair is generated using two large prime numbers. The modulus is formed as the product of these primes, and encryption and decryption are performed using modular exponentiation with public and private exponents, respectively. The underlying hardness assumption is the integer factorization problem, which remains computationally infeasible for sufficiently large key sizes under classical computing conditions. In practical deployments, RSA is commonly integrated into protocols such as TLS/SSL to establish secure channels between microservices. It plays a critical role in authentication and secure key distribution, where a symmetric session key is exchanged securely using the public key mechanism. Due to its deterministic structure, RSA is also employed in digital signatures to ensure message authenticity and non-repudiation. However, optimal security requires large key sizes, typically 2048 bits or higher, which increases computational overhead and impacts latency-sensitive applications.

From a performance perspective, RSA is significantly slower compared to symmetric cryptographic algorithms due to the complexity of modular exponentiation operations. This limitation restricts its use primarily to key exchange rather than bulk data encryption. In microservice architectures, where high-frequency communication occurs between services, excessive reliance on RSA can lead to increased latency and reduced throughput, especially under high-load conditions. A major limitation of RSA arises in the context of quantum computing. The algorithm is fundamentally vulnerable to quantum attacks, particularly through Shor's algorithm, which can efficiently factor large integers in polynomial time. This capability directly compromises the security foundation of RSA, rendering it ineffective in a post-quantum environment. As a result, continued reliance on RSA without transition mechanisms exposes cloud systems to future security risks.

Furthermore, RSA lacks inherent support for cryptographic agility, which is essential in dynamic environments such as cloud microservices. It does not provide built-in mechanisms for seamless algorithm transition or interoperability with emerging PQC schemes. This rigidity complicates migration strategies and necessitates additional frameworks to manage cryptographic evolution. In summary, while RSA remains a critical component of current secure communication infrastructures due to its maturity and widespread support, its computational inefficiency and vulnerability to quantum attacks limit its suitability for future-proof systems. These constraints highlight the necessity for lifecycle-aware transition mechanisms that can gradually replace or augment RSA with quantum-resistant alternatives while maintaining operational continuity.

2.2. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a widely adopted symmetric-key cryptographic algorithm used for secure data encryption in modern computing systems. It operates on fixed-size data blocks of 128 bits and supports key lengths of 128, 192, and 256 bits. AES is based on a substitution–permutation network structure, which involves multiple rounds of transformation including byte substitution, row shifting, column mixing, and key addition. These operations ensure strong diffusion and confusion properties, making AES highly resistant to classical cryptanalytic attacks. In cloud microservices, AES is primarily employed for bulk data encryption due to its high computational efficiency and low latency. It is commonly integrated into secure communication protocols and storage systems to protect data confidentiality during transmission and at rest. Compared to asymmetric algorithms such as RSA, AES offers significantly higher throughput and lower resource consumption, making it suitable for real-time and high-frequency service interactions.

From a performance standpoint, AES benefits from hardware acceleration features available in modern processors, such as AES-NI instructions, which further enhance its execution speed and reduce CPU overhead. This advantage makes AES the preferred choice for encrypting large volumes of data in distributed environments where performance is a critical factor. However, despite its strong security under classical computing assumptions, AES is not entirely immune to quantum threats. Grover's algorithm can reduce the effective key strength of symmetric encryption by approximately half, meaning that AES-128 would offer a security level comparable to 64 bits in a quantum

scenario. Although AES-256 remains relatively secure due to its larger key size, it still does not provide the same level of assurance as dedicated post-quantum cryptographic algorithms.

Another limitation of AES in the context of microservices is its dependence on secure key management. Since it is a symmetric algorithm, both communicating parties must share the same secret key, which introduces challenges in key distribution, storage, and rotation. These challenges become more pronounced in large-scale distributed systems with multiple interacting services. In summary, AES remains a highly efficient and reliable encryption standard for current cloud environments, offering excellent performance and strong classical security. However, its partial vulnerability to quantum attacks and reliance on secure key exchange mechanisms necessitate its integration with post-quantum cryptographic frameworks to ensure long-term security in evolving computational landscapes.

2.3. Post-Quantum Cryptography based on Lattice Algorithms (PQCLA)

Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) represents a class of quantum-resistant cryptographic techniques built upon the hardness of mathematical problems defined over high-dimensional lattices. Unlike classical cryptographic methods such as RSA, which rely on integer factorization, lattice-based schemes are based on problems such as the Learning With Errors (LWE), Ring-LWE, and Shortest Vector Problem (SVP), which are considered computationally intractable even for quantum computers. In PQCLA, encryption and key exchange mechanisms are designed using structured lattice constructions that introduce controlled noise into the system. This noise ensures that recovering the original secret without the correct key becomes computationally infeasible. Lattice-based cryptography supports both encryption and digital signature schemes and is a leading candidate in post-quantum standardization efforts due to its strong security foundation and relatively efficient performance compared to other PQC approaches.

Within cloud microservices, PQCLA can be utilized for secure key exchange and data protection, offering resistance against both classical and quantum adversaries. It enables the development of quantum-resilient communication channels, which is essential for long-term data confidentiality in distributed environments. Additionally, lattice-based schemes often support advanced functionalities such as homomorphic encryption and key encapsulation mechanisms, making them suitable for flexible and secure system design. However, PQCLA introduces certain challenges. The computational complexity associated with lattice operations is higher than that of symmetric encryption algorithms, leading to increased processing time and resource utilization. Moreover, lattice-based schemes typically require larger key sizes and ciphertexts, resulting in higher memory consumption and storage overhead. These factors can impact system performance, particularly in resource-constrained or latency-sensitive microservice environments.

Another limitation is the integration complexity. Existing systems designed around classical cryptography may not be directly compatible with lattice-based primitives, requiring modifications in protocol design and system architecture. This lack of seamless interoperability complicates large-scale deployment. In summary, PQCLA offers strong quantum resistance and is a promising candidate for future secure systems. However, its performance overhead and implementation complexity necessitate careful integration strategies, especially in dynamic cloud microservice environments. These challenges highlight the importance of adaptive frameworks that can incorporate lattice-based cryptography alongside classical methods in a controlled and efficient manner.

2.4. Post-Quantum Cryptography based on Hash Algorithms (PQCHA)

Post-Quantum Cryptography based on Hash Algorithms (PQCHA) refers to a class of quantum-resistant cryptographic techniques that rely on the security properties of cryptographic hash functions. These methods are primarily designed for digital signatures and authentication, leveraging constructions such as Merkle trees and one-time or few-time signature schemes. The security of PQCHA is based on the pre-image resistance, second pre-image resistance, and collision resistance of hash functions, which remain robust even in the presence of quantum adversaries. In PQCHA, signature generation typically involves the use of one-time keys combined through hierarchical tree structures to enable multiple signing operations. Verification is performed using publicly known root hashes, ensuring integrity and authenticity of the message. Since hash functions are well-studied and widely trusted primitives, PQCHA schemes are considered highly secure and conservative choices in the post-quantum cryptographic landscape.

In cloud microservices, PQCHA is particularly suitable for ensuring data integrity, authentication, and secure logging mechanisms. It can be effectively applied in scenarios where strong verification guarantees are required, such as service-to-service authentication, audit trails, and provenance validation. Due to its reliance on simple hash operations, PQCHA avoids complex algebraic structures, making it conceptually straightforward and resistant to many forms of cryptanalysis. However, PQCHA exhibits several practical limitations. One of the primary challenges is the large signature size, which can significantly increase communication overhead in distributed systems. Additionally, many hash-based signature schemes are stateful, requiring careful management of key usage to prevent security breaches. Stateless variants exist, but they often introduce additional computational overhead.

From a performance perspective, PQCHA tends to have higher latency in signature generation and verification compared to classical methods, especially when large tree structures are involved. This can impact real-time microservice interactions where low latency is critical. Furthermore, PQCHA is generally not suitable for encryption or key exchange, limiting its applicability to specific cryptographic functions. In summary, PQCHA provides strong quantum-resistant security grounded in well-established hash function properties, making it a reliable choice for authentication and integrity verification. However, its larger signature sizes, state management requirements, and performance overhead pose challenges for seamless integration into high-performance cloud microservice environments. These constraints highlight the need for hybrid approaches that combine the strengths of different cryptographic paradigms.

Crisp key points about the discussed existing methods are enumerated in Table 1.

Table 1: Existing methods' Key points

Author	Procedure Name	Methodology	Advantages	Limitations
Ron Rivest, Adi Shamir, Leonard Adleman	Rivest–Shamir–Adleman (RSA)	Asymmetric cryptography based on large integer factorization using public and private key pairs	Strong security in classical systems; widely adopted; supports encryption and digital signatures	High computational overhead; large key sizes; vulnerable to quantum attacks (Shor's algorithm)
National Institute of Standards and Technology	Advanced Encryption Standard (AES)	Symmetric key encryption using substitution–permutation network with multiple transformation rounds	High throughput; low latency; efficient for bulk data encryption; hardware acceleration support	Key distribution challenges; reduced security under quantum attacks (Grover's algorithm)
Oded Regev	Post-Quantum Cryptography based on Lattice Algorithms (PQCLA)	Cryptographic schemes based on hard lattice problems such as Learning With Errors (LWE) and Shortest Vector Problem (SVP)	Strong quantum resistance; supports encryption and key exchange; flexible cryptographic constructions	High computational complexity; large key and ciphertext sizes; integration challenges

Ralph Merkle	Post-Quantum Cryptography based on Hash Algorithms (PQCHA)	Hash-based cryptography using Merkle trees and one-time or few-time signature schemes	Strong security assumptions; resistant to quantum attacks; simple and well-understood primitives	Large signature sizes; state management complexity; limited to signatures (not suitable for encryption)

Problem Statement: The widespread use of classical cryptographic algorithms such as RSA and AES in cloud microservices exposes systems to significant security risks due to their vulnerability to quantum computing attacks, thereby threatening long-term data confidentiality and integrity. Existing approaches for integrating post-quantum cryptography rely on static replacement strategies, which are not suitable for dynamic microservice environments and often result in interoperability issues, performance degradation, and lack of seamless transition mechanisms.

In addition, there is no structured lifecycle-aware framework to manage the gradual evolution from classical to hybrid and fully post-quantum cryptographic schemes without disrupting service operations. Furthermore, current systems lack cryptographic provenance and verification capabilities, making it difficult to track algorithm transitions, ensure integrity, prevent downgrade attacks, and establish trust across distributed cloud services.

Research Gap: Despite the growing adoption of post-quantum cryptographic techniques, existing solutions primarily focus on standalone algorithm design rather than their seamless integration into dynamic cloud microservice environments. Most approaches emphasize either lattice-based or hash-based security mechanisms without addressing the challenges of interoperability, scalability, and real-time adaptability across distributed services.

Furthermore, current methodologies lack a structured transition framework that enables gradual migration from classical cryptography to hybrid and fully post-quantum schemes. The absence of lifecycle-aware cryptographic management results in rigid implementations that cannot adapt to varying threat levels, system conditions, or service requirements.

Another significant gap lies in the absence of cryptographic provenance and verification mechanisms. Existing systems do not provide traceability of cryptographic operations, key evolution, or algorithm transitions, making them vulnerable to downgrade attacks, unauthorized substitutions, and integrity violations. This lack of verifiable trust limits the reliability of security frameworks in multi-service cloud environments.

Additionally, limited attention has been given to balancing performance and security in PQC adoption. Many post-quantum approaches introduce considerable computational overhead, increased latency, and storage complexity, without offering strategies to optimize these trade-offs. This gap highlights the need for an integrated framework that ensures quantum resilience while maintaining acceptable performance levels in real-time cloud microservices.

Research Objectives:

- To design a lifecycle-aware cryptographic framework that enables structured transition from classical to hybrid and fully post-quantum cryptographic schemes in cloud microservices.
- To develop a Lifecycle-Aware PQC Transition Engine (LAPTE) for adaptive selection and seamless switching of cryptographic algorithms based on system context and threat levels.
- To implement a Cryptographic Provenance and Verification Engine (CPVE) to ensure traceability, integrity, and validation of cryptographic operations across distributed services.
- To integrate lattice-based and hash-based post-quantum cryptographic techniques with classical methods to achieve a balanced hybrid security model.
- To evaluate the performance of the proposed framework using parameters such as throughput, end-to-end latency, CPU utilization, memory consumption, storage overhead, and security strength.
- To analyze the trade-off between computational efficiency and quantum resistance to ensure practical applicability in real-time cloud microservice environments.

3. BACKGROUND

The integration of post-quantum cryptographic mechanisms into cloud microservices requires a strong

technical foundation that addresses both adaptive cryptographic operation and verifiable security assurance. Modern distributed systems demand flexible cryptographic frameworks capable of evolving in response to emerging threats while maintaining system performance and interoperability. In this context, two critical technical aspects are essential: cryptographic agility with hybrid PQC mechanisms, and cryptographic provenance modelling with verification techniques. These concepts form the basis for enabling controlled cryptographic transition and ensuring trust validation across dynamic service environments.

3.1. Cryptographic Agility and Hybrid PQC Mechanisms

Cryptographic agility refers to the capability of a system to dynamically switch between different cryptographic algorithms without requiring significant architectural modifications. In cloud microservices, where services are independently deployed and frequently updated, agility is crucial to support evolving security requirements. Traditional systems are tightly coupled with specific cryptographic primitives, making it difficult to replace or upgrade algorithms in response to new vulnerabilities. This limitation becomes critical in the transition toward post-quantum cryptography.

Hybrid PQC mechanisms are introduced to address this challenge by combining classical and post-quantum cryptographic algorithms within a unified framework. In such models, both classical and PQC-based keys are generated and utilized together, ensuring backward compatibility while enhancing quantum resistance. For instance, a hybrid key exchange process may involve the simultaneous use of a classical public-key algorithm and a lattice-based Key Encapsulation Mechanism (KEM), where the final shared secret is derived from both components [15].

The implementation of hybrid mechanisms introduces several technical considerations. The size of keys and ciphertexts in PQC schemes is typically larger than classical counterparts, leading to increased communication overhead. Additionally, performing dual cryptographic operations can introduce latency, particularly in real-time microservice interactions. Efficient management of these trade-offs is essential to maintain acceptable system performance.

Another important aspect is the negotiation and selection of cryptographic algorithms between communicating services. Protocols must support flexible configuration to enable services to agree on suitable cryptographic schemes based on capability, policy, and threat level. This requires standardized interfaces and modular cryptographic components that can be integrated without disrupting existing workflows.

Overall, cryptographic agility combined with hybrid PQC mechanisms provides a practical pathway for gradual migration from classical to quantum-resistant security. It ensures continuity of service operations while enabling systems to adapt to future cryptographic requirements.

3.2. Cryptographic Provenance Modelling and Verification Techniques

Cryptographic provenance refers to the ability to track and verify the origin, evolution, and usage of cryptographic operations within a system. In distributed cloud microservices, where multiple services interact across different trust boundaries, maintaining such traceability is essential for ensuring system integrity and accountability. Without provenance mechanisms, it becomes difficult to validate whether correct cryptographic algorithms and keys are being used throughout the system lifecycle. Provenance modelling typically involves the use of structured data representations such as hash chains, Merkle trees, and secure logs. These structures provide tamper-evident properties, ensuring that any modification to recorded cryptographic events can be detected. For example, a sequence of cryptographic operations can be recorded as a hash-linked chain, where each entry depends on the previous one, thereby preserving integrity across the entire sequence.

Verification techniques are applied to validate the correctness and authenticity of cryptographic processes. These include hash-based integrity checks, digital signature verification, and consistency validation across distributed nodes. Such mechanisms enable systems to confirm that cryptographic operations have not been altered or replaced by unauthorized entities. This is particularly important in preventing attacks such as cryptographic downgrade, where weaker algorithms are substituted in place of stronger ones [16].

Key lineage tracking is another critical component of provenance systems. It involves maintaining records of key generation, distribution, usage, rotation, and revocation. Proper tracking ensures that compromised or outdated keys can be identified and replaced in a timely manner. In large-scale microservice

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

environments, automated lineage tracking is necessary to manage the complexity of key lifecycle operations.

In summary, cryptographic provenance modelling and verification techniques provide a robust foundation for establishing trust in distributed systems. By enabling traceability and validation of cryptographic operations, these techniques enhance security, support auditability, and ensure resilience against integrity-related attacks in cloud microservices.

4. PROPOSED METHOD

The proposed Lifecycle-Aware Post-Quantum Cryptography with Provenance Verification (LAPQC) framework introduces a structured and adaptive approach for securing cloud microservices against emerging quantum threats. The method is designed to enable seamless transition from classical cryptographic schemes to hybrid and fully post-quantum mechanisms while maintaining system performance and interoperability. It integrates dynamic cryptographic selection with verifiable trust mechanisms to ensure both operational continuity and security assurance. The framework is composed of two core modules: the Lifecycle-Aware PQC Transition Engine (LAPTE), which manages context-aware cryptographic evolution and algorithm switching, and the Cryptographic Provenance and Verification Engine (CPVE), which ensures traceability, integrity, and validation of cryptographic operations across distributed services. Together, these modules provide a unified solution that addresses the challenges of cryptographic agility, quantum resilience, and trust verification in modern cloud microservice environments.

4.1. Lifecycle-Aware PQC Transition Engine (LAPTE)

The Lifecycle-Aware PQC Transition Engine (LAPTE) is designed to enable controlled and adaptive transition from classical cryptographic schemes to hybrid and fully post-quantum mechanisms within cloud microservices. It introduces a structured lifecycle model that governs the evolution of cryptographic states based on system context, threat levels, and operational requirements. Unlike static replacement approaches, LAPTE supports dynamic selection and seamless switching of cryptographic algorithms, ensuring uninterrupted service execution during transition phases.

Let Sc denotes the Classical State, Sh denotes the Hybrid State, and Sq denotes the Post Quantum state, then the transition engine defines a set of cryptographic states $S = \{Sc, Sh, Sq\}$. The transition between these states is defined as in the following equation.

$$St = f(Ct, Lt, Tt, Rt) \quad \text{Equation (1)}$$

where Ct is the computational load, Lt is the Latency constraints, Tt refers the threat level, and Rt is the resource availability at timestamp t. The function $f(\cdot)$ determines the optimal cryptographic state based on the real-time condition at timestamp t.

Let α, β, γ are the weighting coefficients, then the weighting factor normalization condition is defined as follows.

$$lb(e\alpha \cdot e\beta \cdot e\gamma) = 1 \quad \text{Equation (2).}$$

The computation of decision score D is given in equation 3.

$$D = \alpha \cdot \frac{Tt}{Tmax} + \beta \cdot \left(1 - \frac{Lt}{Lmax}\right) + \gamma \cdot \left(1 - \frac{Ct}{Cmax}\right) \quad \text{Equation (3)}$$

The current cryptographic state is determined by the value of D as follows.

$$St = \begin{cases} Sc \ if \ D < \theta 1 \\ Sh \ if \ \theta 1 \leq D < \theta 2 \\ Sq \ otherwise \end{cases} \quad \text{Equation (4)}$$

where $\theta 1$ and $\theta 2$ are the predefined thresholds.

In the Hybrid state, a combined key is generated using both classical and PQC components. Let Kc be the classical key and Kq be the PQC-derived key. The final session key Kh is computed by equation 5.

$$Kh = H(Kc \| Kq) \quad \text{Equation (5)}$$

where $H(\cdot)$ is a secure cryptographic hash function and $\|$ denotes concatenation. This ensures that the compromise of one component does not weaken the overall security. The latency overhead introduced by hybrid operations is modelled as follows.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

$$Ltotal = (limx \to 0 \frac{(1+xLC)(1+xLq)-1}{x}) + Lcomb \quad \text{Equation (6)}$$

where Lc and Lq represent the latency of classical and PQC operations, and $Lcomb$ denotes the additional cost of key combination and negotiation.

Similarly, the throughput τ under hybrid mode is formulated as in the following equation.

$$\tau h = \left(\frac{1}{\tau c} + \frac{1}{\tau q}\right)^{-1} \quad \text{Equation (7)}$$

where τc and τq are the throughputs of classical and PQC schemes respectively, reflecting the combined processing constraint.

LAPTE also incorporates a cryptographic negotiation mechanism, where each microservice advertises its supported algorithm set Ai. The mutually agreed algorithm $A*$between two services i and j is determined as follows.

$$A* = Ai \cap Aj \quad \text{Equation (8)}$$

The optimal algorithm from $A*$ is then selected based on the decision score D. Overall, the mathematical formulation of LAPTE introduces a novel threshold-driven and context-aware decision framework for systematic and adaptive cryptographic transition, uniquely enabling dynamic selection between classical, hybrid, and post-quantum states while effectively balancing security requirements with performance constraints in cloud microservice environments. The flowchart of the Lifecycle-Aware PQC Transition Engine (LAPTE) module is provided in Figure 1, illustrating the complete operational workflow, including context collection, decision-making, adaptive state selection, cryptographic negotiation, lifecycle-aware key management, execution, and continuous monitoring with re-evaluation for dynamic cryptographic transition.

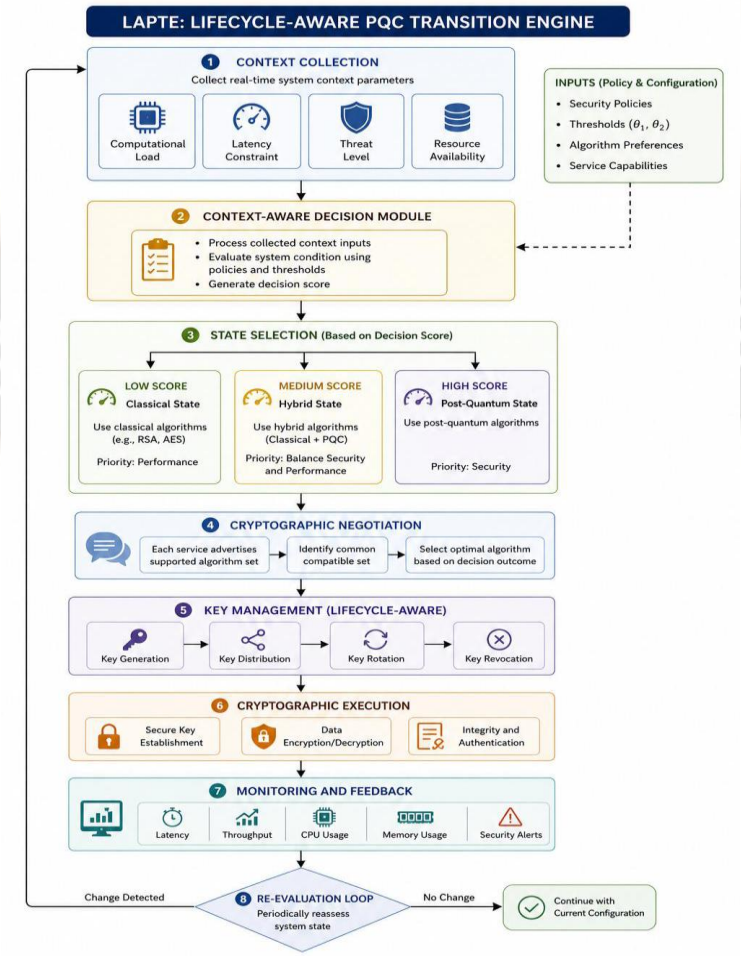

Figure 1: LAPTE Flowchart

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

4.2. Cryptographic Provenance and Verification Engine (CPVE)

The Cryptographic Provenance and Verification Engine (CPVE) is designed to establish a verifiable trust layer within cloud microservices by ensuring traceability, integrity, and authenticity of cryptographic operations. It addresses the critical limitation of existing systems, which lack mechanisms to track the origin and evolution of cryptographic processes across distributed environments. CPVE introduces a structured provenance model that records cryptographic events, key lifecycle transitions, and algorithm selections, thereby enabling continuous validation and auditability.

The engine captures provenance information at each stage of cryptographic execution, including key generation, distribution, usage, rotation, and revocation. Each event is represented as a provenance record containing attributes such as timestamp, service identity, algorithm identifier, key reference, and operation type. These records are organized in a sequential and tamper-evident structure, ensuring that any unauthorized modification can be detected. To maintain integrity, CPVE employs a hash-linked chaining mechanism, where each provenance record is cryptographically bound to the previous record. Let Pi denote the ithprovenance record, and $H(\cdot)$ represent a secure hash function. The linkage is defined as in equation 9.

$$Ci = H(Pi \| Ci-1) \quad \text{Equation (9)}$$

where Ci is the cumulative hash value for the iii-th record. This structure ensures that any alteration in a record disrupts the entire chain, thereby providing strong tamper resistance.

Verification is performed through a combination of hash validation and digital signature mechanisms. Each provenance chain is periodically signed using a service-specific private key, generating a signature σ that can be verified using the corresponding public key as in equation 10.

$$\sigma = Signsk(Cn) \quad \text{Equation (10)}$$

where Cn is the final cumulative hash of the provenance chain. This enables external entities or auditing services to validate the integrity and authenticity of the recorded cryptographic operations.

CPVE also incorporates key lineage tracking, which maintains the relationship between successive keys throughout their lifecycle. Let Ki and $Ki+1$ are consecutive keys; their linkage is defined by equation 11.

$$Ki+1 = f(Ki, \Delta) \quad \text{Equation (11)}$$

where $f(\cdot)$ represents a transformation function and Δ denotes the update parameter, such as time or policy-driven rotation. This ensures traceability of key evolution and facilitates secure key management.

The workflow diagram of CPVE module is provided in Figure 2a and 2b.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

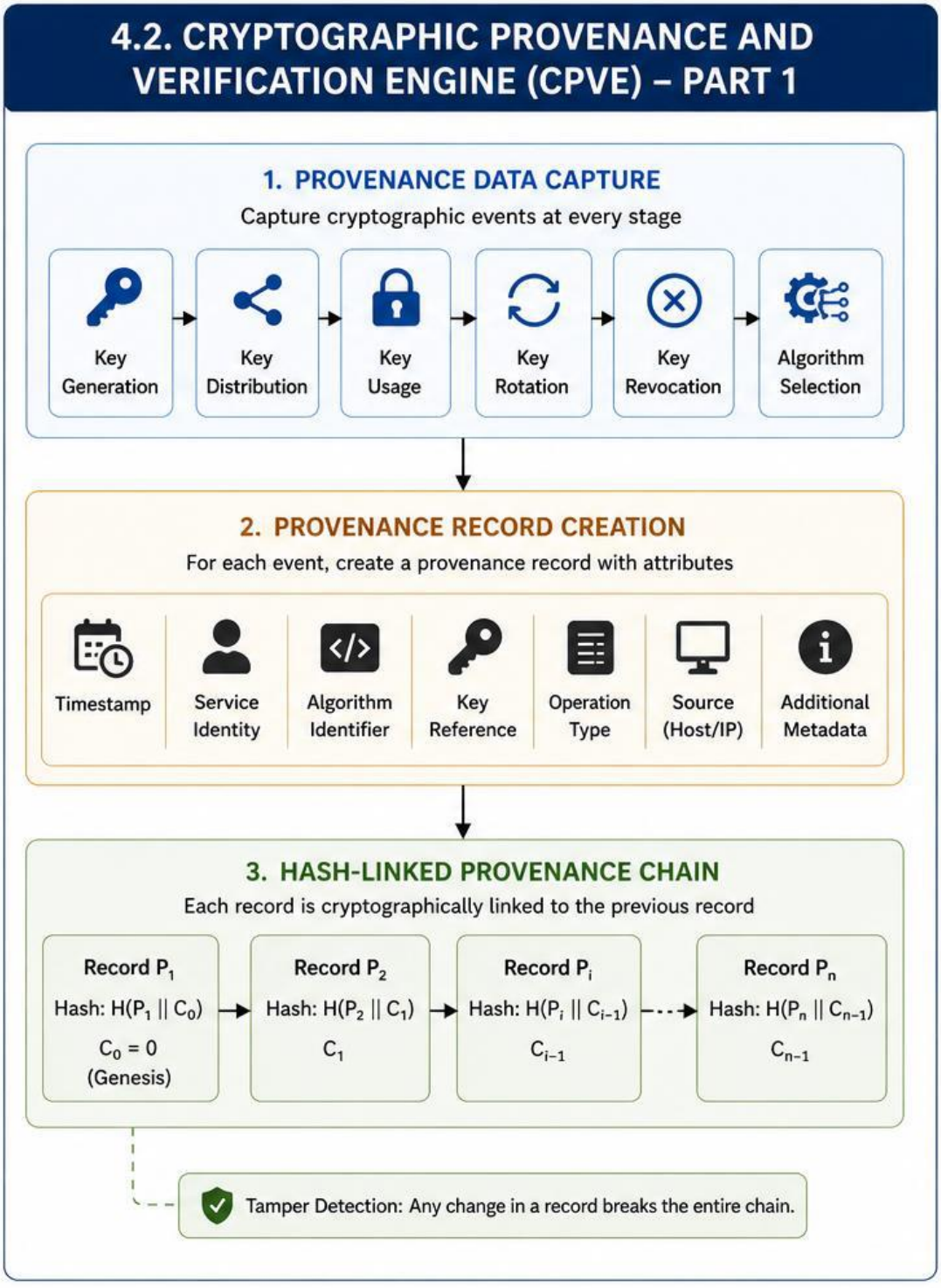

Figure 2(a): CPVE Workflow part 1

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

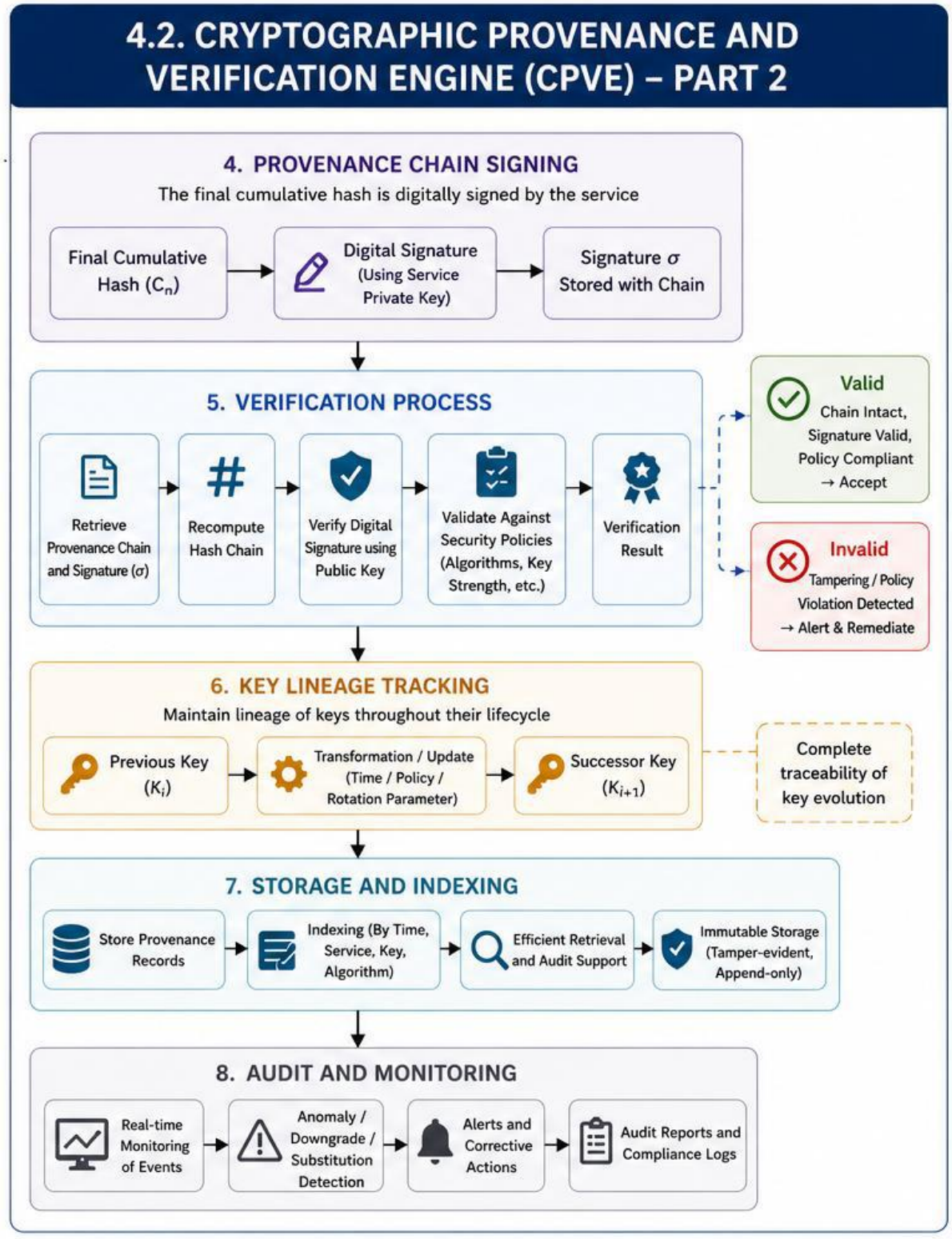

Figure 2(b): CPVE workflow part 2

To mitigate security threats, CPVE includes verification checks against downgrade and substitution attacks. By validating the recorded algorithm identifiers and comparing them with expected security policies, the system ensures that weaker or unauthorized cryptographic schemes are not introduced during operation. Any inconsistency triggers an alert and initiates corrective measures. From the performance perspective, CPVE is optimized to minimize overhead by employing lightweight hash

operations and efficient storage structures. Provenance records are compressed and indexed to support fast retrieval and verification, ensuring scalability in large-scale microservice environments.

Overall, CPVE introduces a novel provenance-driven verification mechanism that enhances trust, accountability, and security in cloud microservices. By combining tamper-evident recording, cryptographic validation, and key lineage tracking, it ensures that all cryptographic operations are transparent, verifiable, and resistant to integrity-based attacks.

4.3. Integrated LAPQC Framework Operation

The proposed Lifecycle-Aware Post-Quantum Cryptography (LAPQC) framework operates through the tight integration of the Lifecycle-Aware PQC Transition Engine (LAPTE) and the Cryptographic Provenance and Verification Engine (CPVE), forming a unified and adaptive security architecture for cloud microservices. The integration ensures that cryptographic decisions are not only dynamically selected based on system conditions but are also continuously validated for integrity, authenticity, and policy compliance. The operational workflow begins with the context collection phase, where system parameters such as computational load, latency constraints, threat level, and resource availability are gathered from the microservice environment. These parameters are processed by LAPTE to compute a decision score, which determines the appropriate cryptographic state among Classical, Hybrid, or Post-Quantum modes. Based on this decision, the system selects the corresponding cryptographic algorithms and initiates secure communication between services through a negotiation mechanism.

Once the cryptographic configuration is established, the execution phase is initiated, where key establishment, data encryption or decryption, and integrity verification are performed using the selected algorithms. At this stage, CPVE is activated to record all cryptographic operations as provenance events. Each event, including key generation, algorithm selection, and data processing, is captured and structured into provenance records containing essential metadata such as timestamps, service identity, and operation details. These records are then organized into a hash-linked provenance chain, ensuring tamper-evident storage of cryptographic activities. The final cumulative hash of the chain is digitally signed, enabling external or internal verification of the entire sequence of operations. During or after execution, the verification module of CPVE validates the integrity of the provenance chain, checks the authenticity of the digital signature, and ensures compliance with predefined security policies.

A key aspect of the integrated framework is the feedback-driven interaction between CPVE and LAPTE. If CPVE detects any anomaly, such as a mismatch in hash values, invalid signatures, or violation of security policies, it generates an alert and signals LAPTE to reassess

the cryptographic configuration. This may result in transitioning to a higher security state, such as switching from Hybrid to Post-Quantum mode, thereby enhancing system resilience against potential threats. In addition, the framework supports continuous monitoring and re-evaluation, where performance metrics and security events are periodically analyzed. This enables the system to adapt dynamically to changing conditions, ensuring optimal balance between performance efficiency and security strength. The integration of decision-making, execution, provenance recording, and verification creates a closed-loop system that maintains both operational continuity and trust assurance.

Overall, the integrated LAPQC framework introduces a novel synergy between adaptive cryptographic transition and provenance-driven verification, enabling secure, transparent, and resilient communication in cloud microservice environments. This unified operation not only facilitates smooth migration toward post-quantum cryptography but also ensures that every cryptographic action is traceable, verifiable, and aligned with system security policies.

5. Experimental Setup

The experimental setup is designed to evaluate the performance and effectiveness of the proposed Lifecycle-Aware Post-Quantum Cryptography (LAPQC) framework in a cloud microservice environment. The setup focuses on analyzing the trade-off between computational efficiency and quantum-resilient security by comparing classical, post-quantum, and hybrid cryptographic schemes under real-time workload conditions. The implementation is carried out using a distributed microservice architecture, where multiple services communicate securely using different cryptographic configurations. The system is developed using C++ 23.0 [17] with a Visual Studio IDE [18]. The experimental platform consists of a local machine

equipped with an Intel Core i7 (14th Generation) processor and 16 GB RAM, which handles orchestration, configuration, and user interface operations. Computational workloads are offloaded to a remote server infrastructure (i2k2 server) [19], which provides higher processing capability and acts as an execution layer for cryptographic operations. The i2k2 server is interfaced with IBM Qiskit [20] to enable quantum-aware processing and simulation of post-quantum cryptographic components.

Each microservice is configured to dynamically select cryptographic algorithms through the LAPTE module, while all cryptographic operations are recorded and verified using the CPVE module. The interaction between local and remote environments follows a structured request-response mechanism, ensuring efficient task distribution and result retrieval. The evaluation considers four categories of cryptographic methods: Rivest–Shamir–Adleman (RSA), Advanced Encryption Standard (AES), Post-Quantum Cryptography based on Lattice Algorithms (PQCLA), and Post-Quantum Cryptography based on Hash Algorithms (PQCHA), along with the proposed hybrid approach. These methods are executed under identical conditions to ensure fair comparison.

Performance is measured using standardized parameters including throughput, end-to-end latency, CPU utilization, memory consumption, storage overhead, and security strength. The experiments are conducted across multiple timestamps under typical workload conditions, where each timestamp represents a consistent execution scenario with varying system dynamics. The communication between microservices involves secure key establishment, data encryption and decryption, and integrity verification processes. For hybrid configurations, both classical and post-quantum components are executed in parallel, and the combined output is used for secure communication. The CPVE module continuously logs cryptographic events, constructs provenance chains, and performs verification checks to ensure integrity and policy compliance during execution.

To simulate realistic operational conditions, system parameters such as computational load, latency constraints, and resource availability are varied within controlled limits. The threat level is also adjusted to trigger different cryptographic states within the LAPTE module, enabling evaluation of adaptive transition behavior. All experiments are repeated multiple times, and average values are recorded to ensure consistency and reliability. The experimental setup provides a comprehensive environment to validate the proposed LAPQC framework, demonstrating its ability to achieve adaptive cryptographic transition, maintain system performance, and ensure verifiable security in cloud microservice architectures. Hardware and software configurations are enumerated in Table 2.

Category	**Specification**
Processor	Intel Core i7 (12th Generation)
RAM	16 GB DDR4
Storage	1 TB NVMe M.2 SSD
Operating System	Windows 10 / 11 (64-bit)
Development Environment	Visual Studio (C++ 23.0)
Programming Language	C++ 23.0
Architecture	Distributed Cloud Microservice Architecture
Local System Role	UI, Configuration, Orchestration
Remote Server	i2k2 Server
Server Role	Cryptographic Processing and Execution
Quantum Integration	IBM Qiskit (via i2k2 server)
Communication Model	Request–Response (Local ↔ Server)
Cryptographic Methods	RSA, AES, PQCLA, PQCHA, Hybrid PQC
Evaluation Parameters	Throughput, Latency, CPU Utilization, Memory Usage, Storage Overhead, Security Strength

Table 2: LAPQC Hardware and Software configurations

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

6. RESULTS AND DISCUSSIONS

The performance of the proposed Lifecycle-Aware Post-Quantum Cryptography (LAPQC) framework is evaluated using multiple cryptographic schemes, including Rivest–Shamir–Adleman (RSA), Advanced Encryption Standard (AES), Post-Quantum Cryptography based on Lattice Algorithms (PQCLA), Post-Quantum Cryptography based on Hash Algorithms (PQCHA), and the proposed hybrid approach (PQCHY). The evaluation is conducted across ten timestamps under consistent real-time workload conditions. Key performance parameters such as throughput and end-to-end delay are considered to analyze system efficiency and responsiveness in cloud microservice environments. The observed values provide a comparative understanding of classical, post-quantum, and hybrid cryptographic methods in terms of performance and computational overhead.

6.1. Throughput

Throughput represents the number of requests processed per second and is a critical parameter for evaluating the efficiency of cryptographic operations in cloud microservice environments. It reflects the ability of the system to handle high volumes of secure communication without performance degradation. In this analysis, throughput is measured across ten timestamps for all considered cryptographic methods, including RSA, AES, PQCLA, PQCHA, and the proposed hybrid approach (PQCHY), under consistent workload conditions. The observed values provide insight into the computational efficiency and scalability of each method, particularly in balancing performance with enhanced security requirements. Observed throughput values are provided in Table 3.

Throughput (Requests/Second)					
Timestamp					**PQCHY**
1					952
2	747	1061 1043	968	715	928
3	759		934	708	937
4	742	1040 1082 1079	955	655	937
5	791	1088	941	683	923
6	735		971	710	888
7	772		953	695	947
8	775	1037 1090 1035	975	663	891
9	761		915	639	921
10	762		946	660	895

Table 3: Throughput

The throughput performance of the evaluated cryptographic methods is analyzed based on the number of requests processed per second across ten timestamps. The observed values indicate that Advanced Encryption Standard (AES) consistently achieves the highest throughput, ranging approximately from 1035 to 1090 requests per second, due to its efficient symmetric encryption mechanism. Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) demonstrates strong performance among PQC methods, maintaining throughput

values in the range of 915 to 975 requests per second. The proposed hybrid approach (PQCHY) achieves comparable performance, with values between 888 and 952 requests per second, indicating its ability to balance security and efficiency. In contrast, Rivest–Shamir–Adleman (RSA) exhibits moderate throughput, varying between 735 and 791 requests per second, reflecting its higher computational overhead. Post-Quantum Cryptography based on Hash Algorithms (PQCHA) records the lowest throughput, ranging from 639 to 715 requests per second, due to increased processing complexity.

Overall, the results demonstrate that the proposed hybrid approach maintains throughput close to PQCLA while significantly outperforming PQCHA, thereby providing an effective trade-off between computational efficiency and quantum-resistant security. The comparison graph of discussed methods in context with throughput is provided in Figure 3.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

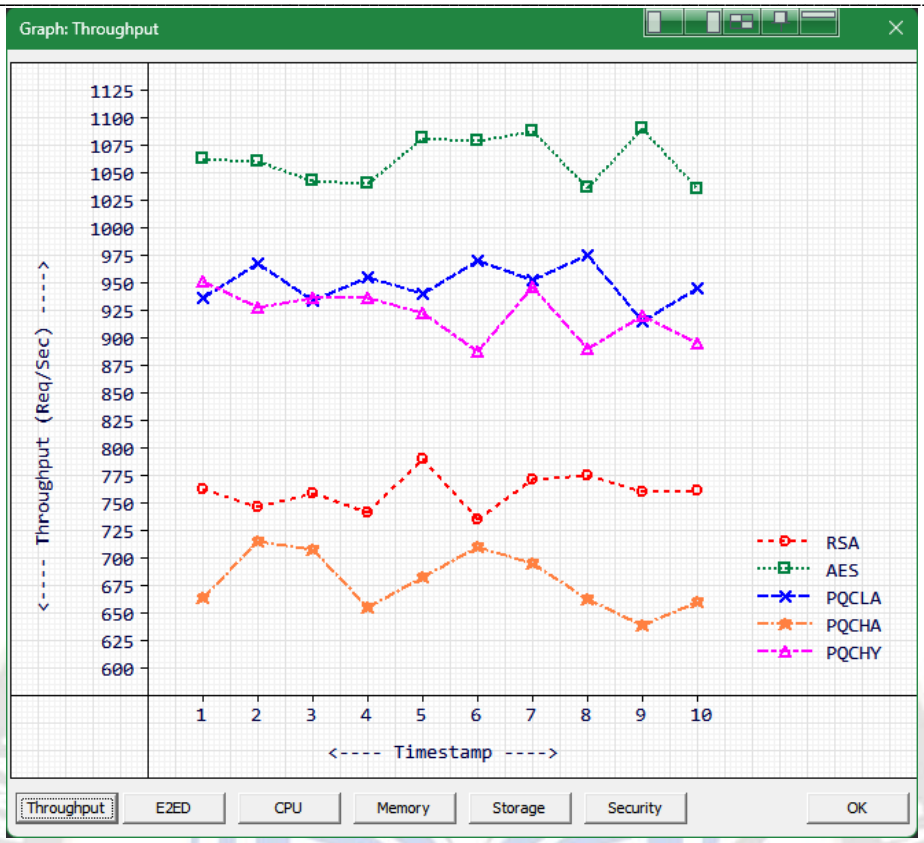

Table 3: Throughput

6.2. End-to-End Delay

End-to-end delay represents the total time required for a request to be processed from initiation to completion, including key establishment, encryption, transmission, and decryption. It is a critical parameter for evaluating the responsiveness of cryptographic operations in cloud microservice environments. In this analysis, end-to-end delay is measured in milliseconds across ten timestamps for all considered cryptographic methods, including RSA, AES, PQCLA, PQCHA, and the proposed hybrid approach (PQCHY). The observed values provide insight into the latency overhead introduced by each method, particularly in balancing security

requirements with real-time performance constraints. Measured End-to-End Delay values are provided in Table 4 and the comparison graph is provided in Figure 4.

End-to-End delay (mS)					
Timestamp				**PQCHA**	**PQCHY**
1	49	28	37	68	41
2	49	29	35	66	40
3	44	24	37	63	30
4	47	27	29	72	35
5	52	20	36	65	34
6	48	20	34	70	40
7	55	23	28	65	33
8	46	30	30	68	37
9	51	22	35	67	33
10	48	20	38	67	38

Table 4: End-to-End delay

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

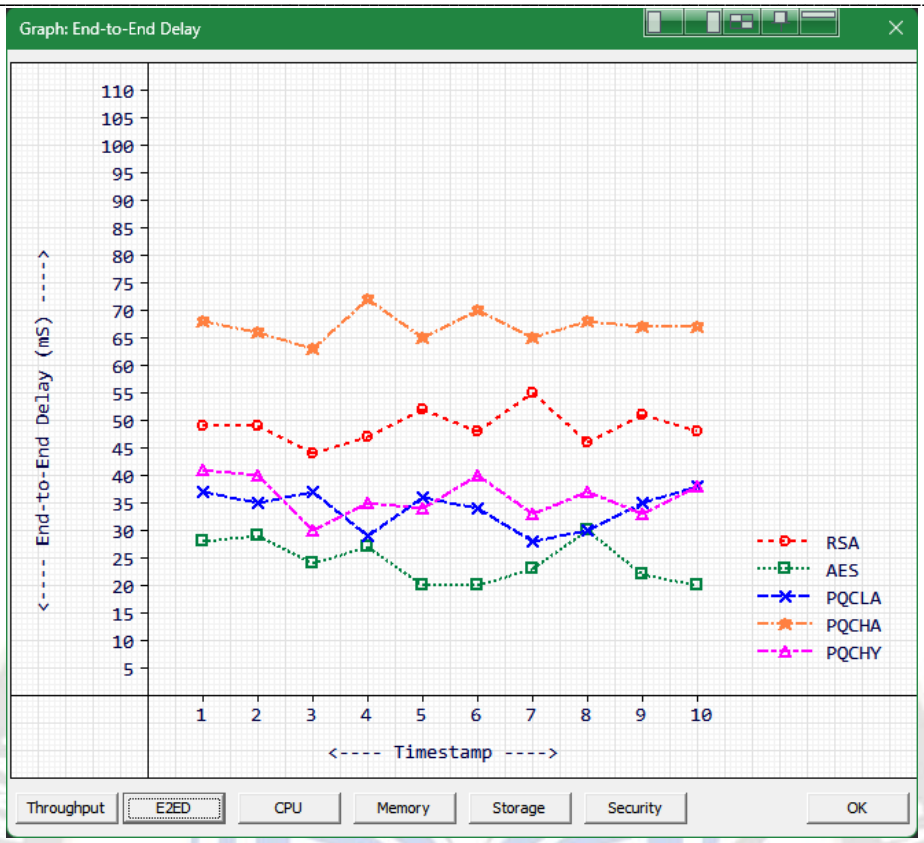

Figure 4: End-to-End Delay

The end-to-end delay of the evaluated cryptographic methods is analyzed based on the total processing time measured in milliseconds across ten timestamps. The observed values indicate that Advanced Encryption Standard (AES) consistently achieves the lowest delay, ranging approximately from 20 to 30 mS, due to its efficient symmetric encryption mechanism. Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) demonstrates moderate delay values between 28 and 38 mS, reflecting a balanced trade-off between computational

complexity and performance. The proposed hybrid approach (PQCHY) maintains comparable latency, with values ranging from 30 to 41 mS, indicating its ability to integrate quantum-resistant mechanisms without significant delay overhead.

In contrast, Rivest–Shamir–Adleman (RSA) exhibits higher delay values between 44 and 55 mS due to the computational cost of asymmetric operations. Post-Quantum Cryptography based on Hash Algorithms (PQCHA) records the highest delay, ranging from 63 to 72 mS, highlighting its increased processing overhead. Overall, the results demonstrate that the proposed hybrid approach achieves lower latency than standalone PQC methods while maintaining acceptable performance compared to classical algorithms, thereby ensuring efficient and responsive secure communication in cloud microservice environments.

6.3. CPU Utilization

CPU utilization represents the percentage of processor resources consumed during cryptographic operations and is a key indicator of computational efficiency in cloud microservice environments. It reflects the processing overhead introduced by different cryptographic schemes during key generation, encryption, decryption, and verification processes. In this analysis, CPU utilization is measured across ten timestamps for all considered methods, including RSA, AES, PQCLA, PQCHA, and the proposed hybrid approach (PQCHY), under consistent workload conditions. The observed values provide insight into the computational demand of each method and their impact on system performance, particularly in scenarios requiring scalable and real-time secure communication. Metered CPU utilizations of discussed methods during the experiment are provided in Table 5.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

CPU Utilization (%)					
Timestamp				**PQCHA**	**PQCHY**
1				89	56
2	61	46	48	81	54
3	74	41	48	89	52
4	75	48	44	71	72
5	62	43	59	66	55
6	63	57	47	77	57
7	61	55	62	86	66
8	65	37	62	80	63
9	75	53	48	83	64
10	63	56	54	87	63

Table 5: CPU Utilization

The CPU utilization of the evaluated cryptographic methods is analyzed based on the percentage of processor resources consumed across ten timestamps. The observed values indicate that Advanced Encryption Standard (AES) consistently exhibits the lowest CPU utilization, ranging approximately from 35% to 57%, highlighting its computational efficiency.

Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) demonstrates moderate CPU usage, varying between 44% and 62%, indicating a balanced computational demand. The proposed hybrid approach (PQCHY) shows slightly higher utilization, ranging from 52% to 72%, due to the combined execution of classical and post-quantum components. In contrast, Rivest–Shamir–Adleman (RSA) exhibits relatively higher CPU utilization between 61% and 80%, reflecting the computational overhead of asymmetric operations. Post-Quantum Cryptography based on Hash Algorithms (PQCHA) records the highest CPU usage, ranging from 66% to 89%, due to its intensive processing requirements.

Overall, the results demonstrate that the proposed hybrid approach maintains moderate CPU utilization while providing enhanced security, achieving a practical balance between computational efficiency and quantum-resistant performance. The comparison graph for CPU utilization is provided in Figure 5.

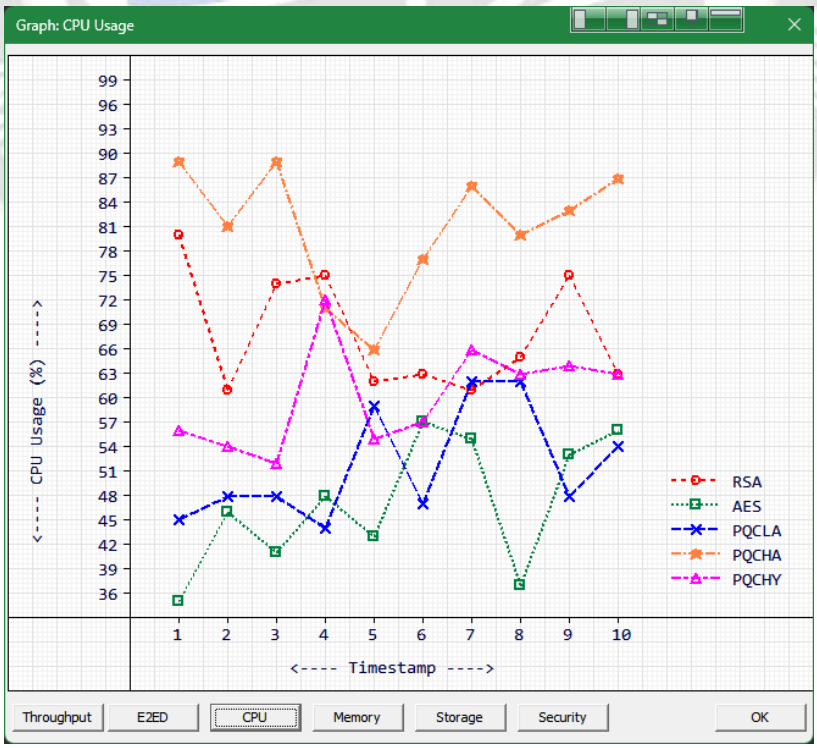

Figure 5: CPU Utilization

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

6.4. Memory Consumption

Memory consumption represents the amount of system memory utilized during cryptographic operations and is an important parameter for evaluating resource efficiency in cloud microservice environments. It reflects the memory overhead associated with key storage, encryption processes, and intermediate computations. In this analysis, memory consumption is measured across ten timestamps for all considered cryptographic methods, including RSA, AES, PQCLA, PQCHA, and the proposed hybrid approach (PQCHY), under consistent

workload conditions. The observed values provide insight into the resource requirements of each method and their suitability for scalable and resource-constrained environments. Measured memory consumptions of the discussed methods during different timestamps are listed in table 6, and the comparison graph is given in Figure 6.

Memory Consumption (MB)					
Timestamp				**PQCHA**	**PQCHY**
1				281	252
2	226		245	298	264
3	209		247	303	267
4	231	183 164 158 177	239	285	279
5	215		246	281	271
6	227		259	304	267
7	204		242	309	266
8	232	155 154	255	302	280
9	207		230	303	270
10	232	153	261	285	255

Table 6: Memory Consumption

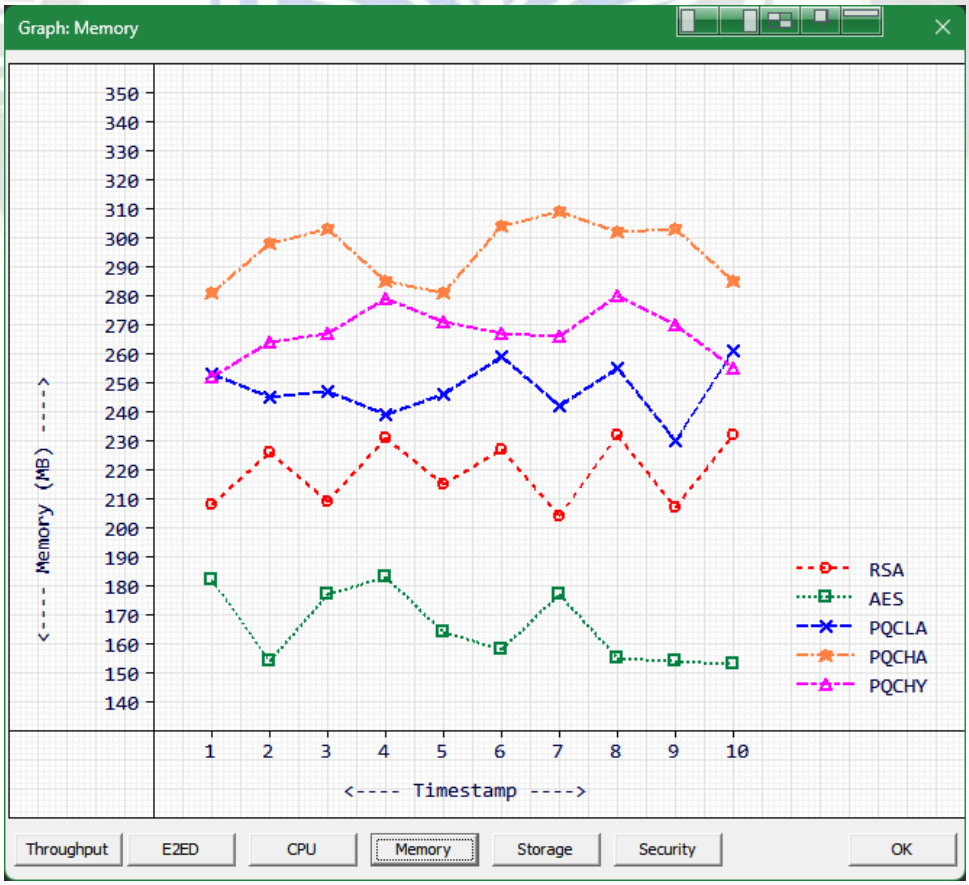

Figure 6: Memory Consumption

The memory consumption of the evaluated cryptographic methods is analyzed based on the amount of memory utilized in megabytes across ten timestamps. The observed values indicate that Advanced Encryption Standard (AES) consistently requires the least memory, ranging approximately from 153 MB to 183 MB, demonstrating its efficiency in resource

utilization. Rivest–Shamir–Adleman (RSA) exhibits moderate memory usage between 204 MB and 232 MB, reflecting the overhead associated with key handling and asymmetric operations. Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) shows higher memory consumption, varying from 230 MB to 261 MB, due to larger key sizes and complex computations.

The proposed hybrid approach (PQCHY) records memory usage in the range of 252 MB to 280 MB, slightly higher than PQCLA, as it combines both classical and post-quantum components. In contrast, Post-Quantum Cryptography based on Hash Algorithms (PQCHA) demonstrates the highest memory consumption, ranging from 281 MB to 309 MB, due to its large signature sizes and storage requirements. Overall, the results indicate that the proposed hybrid approach maintains moderate memory usage while providing enhanced security, achieving a practical balance between resource utilization and quantum-resistant capabilities.

6.5. Storage Overhead

Storage overhead represents the additional storage required to accommodate cryptographic elements such as keys, ciphertexts, signatures, and provenance records during secure communication. It is an important parameter for evaluating the scalability and practicality of cryptographic schemes in cloud microservice environments. In this analysis, storage overhead is measured across ten timestamps for all considered methods, including RSA, AES, PQCLA, PQCHA, and the proposed hybrid approach (PQCHY), under consistent workload conditions. The observed values provide insight into the storage requirements of each method and their impact on system efficiency, particularly in large-scale distributed deployments. Measured storage overheads during the experiment are provided in Table 7.

Storage Overhead (MB)					
Timestamp	**RSA**	**AES**	**PQCLA**	**PQCHA**	**PQCHY**
1	57	28	82	114	95
2	62	30	81	119	95
3	60	21	90	112	90
4	57	17	79	122	88
5	56	21	88	119	98
6	62	28	87	123	89
7	56	20	92	124	97
8	53	27	93	120	100
9	53	26	93	125	99
10	53	27	82	125	102

Table 7: Storage Overhead

The storage overhead of the evaluated cryptographic methods is analyzed based on the additional storage required in megabytes across ten timestamps. The observed values indicate that Advanced Encryption Standard (AES) consistently incurs the lowest storage overhead, ranging approximately from 17 MB to 30 MB, due to its compact key and ciphertext structure.

Rivest–Shamir–Adleman (RSA) exhibits moderate storage requirements between 53 MB and 62 MB, reflecting the overhead associated with key storage and asymmetric operations. Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) shows higher storage usage, varying from 79 MB to 93 MB, due to larger key and ciphertext sizes inherent to lattice-based schemes. The proposed hybrid approach (PQCHY) records storage overhead in the range of 88 MB to 102 MB, slightly higher than PQCLA, as it integrates both classical and post-quantum components. In contrast, Post-Quantum Cryptography based on Hash Algorithms (PQCHA) demonstrates the highest storage overhead, ranging from 112 MB to 125 MB, due to large signature sizes and associated metadata.

Overall, the results indicate that the proposed hybrid approach maintains moderate storage overhead while providing enhanced quantum-resistant security, achieving a balanced trade-off between storage

efficiency and cryptographic strength in cloud microservice environments. The storage overhead comparison graph is provided in Figure 7.

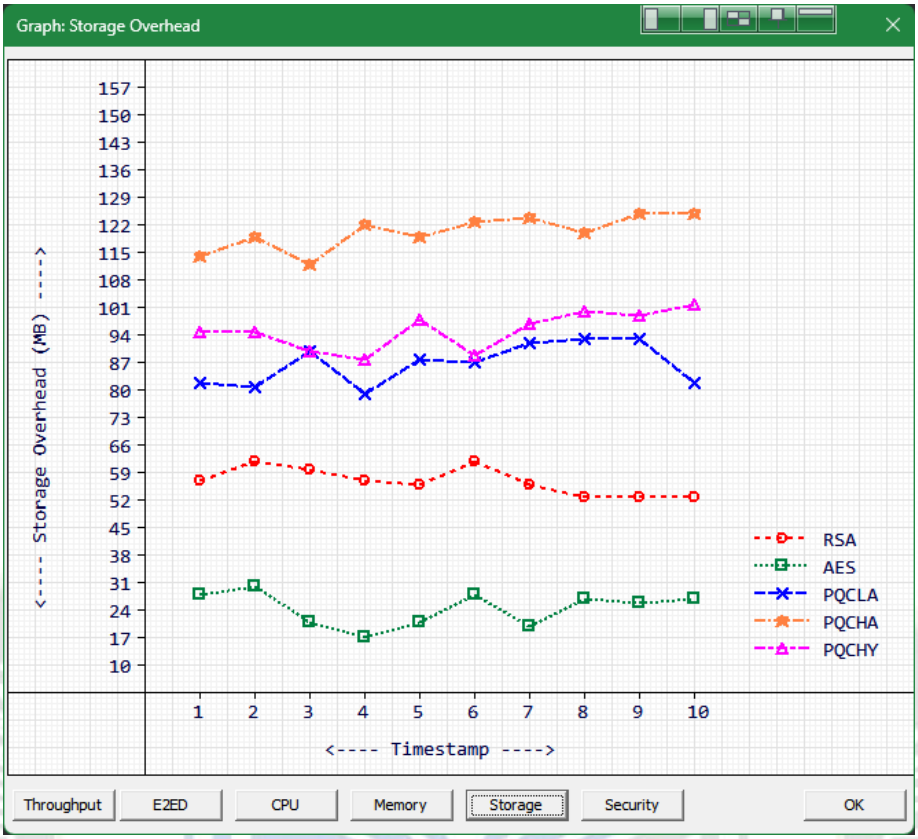

Figure 7: Storage Overhead

6.6. Security (Tamper Detection Rate)

Security strength represents the level of resistance offered by a cryptographic method against potential attacks, particularly in the context of emerging quantum threats. It reflects the effective robustness of the algorithm based on key size, computational hardness, and resistance to both classical and quantum cryptanalysis. In this analysis, security strength is evaluated comparatively for all considered methods, including RSA, AES, PQCLA, PQCHA, and the proposed hybrid approach (PQCHY), under consistent conditions. The assessment provides insight into the ability of each method to ensure long-term confidentiality, integrity, and resilience in cloud microservice environments. Measured security scores of the discussed methods during different timestamps are provided in Table 8 and the comparison graph is given in Figure 8.

Security: Tamper Detection Rate (%)					
Timestamp					**PQCHY**
1	78	82	92.0588	94.1176	98
2	78.3529	82.1765	92.2941 92	94.2941 94.2353	98
3	78.3529	82.1765			98.0588
4	78	82.2353	92	94.2353	98.1765
5	78	82.1176	92.1765	94	98.1765
6	78.1765	82.0588	92.0588 92.1765	94.2941 94.0588	98
7	78.2353	82			98.3529
8	78.3529	82.2941 82.2941	92.1176 92.0588	94.2941 94.3529	98.2941
9	78.0588	82.2941	92	94.1765	98.2941
10	78.1176				98.2941

Table 8: Security

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 July 2026* ***Revised:*** *12 August 2026* ***Accepted:*** *08 September 2026*

Figure 8: Security

The security strength of the evaluated cryptographic methods is analyzed using the tamper detection rate (%), which represents the ability of each method to accurately identify unauthorized modifications in transmitted or stored data. This parameter is of primary

importance in the proposed LAPQC framework, as it directly reflects the effectiveness of the Cryptographic Provenance and Verification Engine (CPVE) in ensuring integrity, traceability, and trust across microservice interactions. The observed values clearly indicate that the proposed hybrid approach (PQCHY) consistently achieves the highest tamper detection rate, maintaining values in the range of 98% to 98.35% across all timestamps. This superior performance is attributed to the integration of provenance-driven verification mechanisms, where every cryptographic operation is recorded, hash-linked, and validated. The combination of classical and post-quantum components further strengthens detection capability, ensuring that any deviation in cryptographic execution or data integrity is immediately identified.

In comparison, Post-Quantum Cryptography based on Hash Algorithms (PQCHA) demonstrates strong detection performance, with values between 94% and 94.35%, due to its inherent reliance on hash-based integrity checks. However, the absence of structured provenance tracking limits its ability to achieve the higher detection accuracy observed in PQCHY. Similarly, Post-Quantum Cryptography based on Lattice Algorithms (PQCLA) records detection rates in the range of 92% to 92.29%, providing reliable security but lacking the additional verification layers present in the proposed framework. Classical methods such as AES and RSA exhibit comparatively lower detection rates, approximately 82% and 78%, respectively. These methods primarily focus on confidentiality and do not inherently provide strong mechanisms for continuous integrity validation or cryptographic traceability. As a result, their ability to detect sophisticated tampering or substitution attacks is limited in distributed environments.

A key insight from the results is the role of provenance-based verification in enhancing tamper detection. The CPVE module ensures that every cryptographic event is recorded in a tamper-evident structure, enabling verification not only of data integrity but also of the correctness of the cryptographic process itself. This multi-layer verification approach significantly improves detection accuracy compared to standalone cryptographic schemes. Furthermore, the consistency of detection rates across all timestamps demonstrates the

stability and reliability of the proposed framework under varying workload conditions. Unlike other methods where detection capability remains static, the integration with LAPTE allows the system to adapt its cryptographic strength dynamically based on threat levels, indirectly reinforcing tamper detection effectiveness.

Overall, the results confirm that the proposed LAPQC framework achieves a substantial improvement in security strength, with the hybrid PQC approach delivering near-optimal tamper detection performance. This establishes the framework as a robust solution for ensuring data integrity, preventing unauthorized modifications, and maintaining trust in cloud microservice environments, which is the primary objective of this proposed work.

7. CONCLUSION

This work presents the Lifecycle-Aware Post-Quantum Cryptography with Provenance Verification (LAPQC) framework for securing cloud microservices against emerging quantum

threats. The proposed method introduces a novel lifecycle-aware cryptographic transition mechanism (LAPTE) combined with a provenance-driven verification engine (CPVE), enabling dynamic selection of cryptographic states along with continuous integrity validation. Unlike existing approaches that rely on static replacement of cryptographic schemes, the proposed framework integrates threshold-driven decision-making, hybrid PQC execution, and tamper-evident provenance tracking, establishing a unified and adaptive security model. The experimental evaluation demonstrates that the proposed hybrid approach achieves a balanced trade-off between performance and security, maintaining throughput close to classical methods while controlling latency and resource overhead. Most importantly, the framework achieves significantly enhanced tamper detection capability (~98%), validating the effectiveness of provenance-based verification in ensuring data integrity and preventing unauthorized modifications. The results confirm that the integration of adaptive cryptographic transition with verifiable trust mechanisms provides a scalable and reliable solution for next-generation cloud microservice environments. Overall, the proposed LAPQC framework successfully addresses key challenges in post-quantum adoption, including interoperability, performance degradation, and lack of traceability, thereby establishing a strong foundation for quantum-resilient and trustworthy distributed systems. As future work, the framework can be extended to incorporate real-time threat intelligence and AI-driven decision models for further optimizing cryptographic selection, along with exploration of lightweight PQC schemes to reduce computational overhead in resource-constrained environments.

REFERENCES

[1] B. M. Harve et al., "The Cloud-Native Revolution: Microservices in a Cloud-Driven World," in 2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), Bali, Indonesia, 2024, pp. 1043–1048, doi: https://doi.org/10.1109/ICICYTA64807.2024.10913359

[2] M. Matias, E. Ferreira, N. Mateus-Coelho, O. Ribeiro, and L. Ferreira, "Evaluating Effectiveness and Security in Microservices Architecture," Procedia Computer Science, vol. 237, pp. 626–636, 2024, doi: https://doi.org/10.1016/j.procs.2024.05.148

[3] M. Kumar and B. Mondal, "Study on Implementation of Shor's Factorization Algorithm on Quantum Computer," SN Computer Science, vol. 5, p. 413, 2024, doi: https://doi.org/10.1007/s42979-024-02771-y

[4] S. Mandal, R. Anand, M. Rahman et al., "Implementing Grover's on AES-based AEAD schemes," Scientific Reports, vol. 14, p. 21105, 2024, doi: https://doi.org/10.1038/s41598-024-69188-8.

[5] N. R. Reddy, S. Suryadevara, K. G. R. Reddy et al., "Quantum secured blockchain framework for enhancing post quantum data security," Scientific Reports, vol. 15, p. 31048, 2025, doi: https://doi.org/10.1038/s41598-025-16315-8

[6] E. Fathalla and M. Azab, "Beyond Classical Cryptography: A Systematic Review of Post-Quantum Hash-Based Signature Schemes, Security, and Optimizations," IEEE Access, vol. 12, pp. 175969–175987, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3485602.

[7] G. D. Schwarz, P. Heltweg, and D. Riehle, "Balancing technology heterogeneity in microservice architectures," Empirical Software

Engineering, vol. 30, p. 127, 2025, doi: https://doi.org/10.1007/s10664-025-10684-4.

[8] R. Campbell, "Enterprise Migration to Post-Quantum Cryptography: Timeline Analysis and Strategic Frameworks," Computers, vol. 15, no. 1, p. 9, 2026, doi:

https://doi.org/10.3390/computers15010009.

[9] G. Niu, "Evaluation of Blockchain-Based Tracking and Tracing System With Uncertain Information: A Multi-Criteria Decision-Making Approach," IEEE Access, vol. 13, pp. 40795–40812, 2025, doi: https://doi.org/10.1109/ACCESS.2025.3546275.

[10] A. El Akhdar, C. Baidada, A. Kartit, M. Hanine, C. O. García, R. G. Lara, and I. Ashraf, "Exploring the Potential of Microservices in Internet of Things: A Systematic Review of Security and Prospects," Sensors, vol. 24, no. 20, p. 6771, 2024, doi:

https://doi.org/10.3390/s24206771

[11] A. C. B. Vassoler, G. B. M. Fernandes, L. R. Kieslich, E. A. Bezerra, and H. Pettenghi, "A New Strategy to Design Reconfigurable Rivest–Shamir–Adleman (RSA) Accelerators," International Journal of Circuit Theory and Applications, pp. 1–12, 2026, doi: https://doi.org/10.1002/cta.70344.

[12] J. Z. E. Basco, A. F. Romualdo, J. A. Saluta, C. S. Santiago, and R. M. Marfil, "Comparison of Data Encryption Standard (DES) and Advanced Encryption Standard (AES) in Security Issues of Cloud Computing: A Literature Review," in Innovations in Information and Decision Sciences, V. Bhateja, M. Dey, and R. Senkerik, Eds., Smart Innovation, Systems and Technologies, vol. 422, Singapore: Springer, 2025, doi: https://doi.org/10.1007/978-981-96-0147-9_16

[13] H. Nguyen, S. Huda, Y. Nogami, and T. T. Nguyen, "Security in Post-Quantum Era: A Comprehensive Survey on Lattice-Based Algorithms," IEEE Access, vol. 13, pp. 89003–89024, 2025, doi: https://doi.org/10.1109/ACCESS.2025.3571307

[14] K. Algazy, K. Sakan, A. Khompysh, and D. Dyusenbayev, "Development of a New Post-Quantum Digital Signature Algorithm: Syrga-1," Computers, vol. 13, no. 1, p. 26, 2024, doi: https://doi.org/10.3390/computers13010026

[15] M. Aljamal, B. S. Khassawneh, A. Alsarhan, S. Okour, L. A. Almusfar, B. F. AlThani, and W. Aldossary, "A Novel Dual-Layer Quantum-Resilient Encryption Strategy for UAV–Cloud Communication Using Adaptive Lightweight Ciphers and Hybrid ECC–PQC," Computers, vol. 15, no. 2, p. 101, 2026, doi: https://doi.org/10.3390/computers15020101.

[16] K. Balan, R. Learney, and T. Wood, "A Framework for Cryptographic Verifiability of End-to-End AI Pipelines," in Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics (IWSPA '25), New York, NY, USA: Association for Computing Machinery, 2025, pp. 49–59, doi: https://doi.org/10.1145/3716815.3729011
https://www.geeksforgeeks.org/cpp/cpp-23-standard/
https://visualstudio.microsoft.com/
https://i2k2.com/
https://www.ibm.com/quantum/qiskit