Journal of Information Systems Engineering and Management
2025, 10(9s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Performance Evaluation of Post-Quantum Cryptography: A Comprehensive Framework for Experimental Analysis

Mrs. Prajakta Pote[1], Dr. Rajesh Bansode[2]
[1]Research Scholar, Dept. of IT, Thakur College of Engg & Tech, Mumbai University, Maharashtra, India
[2]Professor, Dept. of IT, Thakur College of Engg & Tech, Mumbai University, Maharashtra, India
Corresponding email: praj.pote1@gmail.com

ARTICLE INFO

Received: 01 Nov 2024
Revised: 16 Dec 2024
Accepted: 05 Jan 2025

ABSTRACT

Post-quantum cryptography (PQC) is a critical area of research aimed at addressing the threat quantum computing poses to traditional cryptographic systems. It focuses on evaluating the efficiency, security, and practical performance of PQC algorithms through experimental analysis. This research supports the development of optimized cryptographic solutions, contributes to standardization efforts, and ensures secure and efficient implementations for a wide range of applications. Despite progress in PQC, gaps still exist in practical performance evaluations, particularly in resource-constrained environments. There is a lack of standardized comparisons, hindering the selection of suitable algorithms for specific use cases like IoT and cloud systems. Furthermore, research on scalability, integration challenges, and the trade-offs between security and efficiency remains insufficient. The methodology involves creating a comprehensive framework to evaluate various PQC algorithms, including lattice-based (e.g., Kyber, Dilithium), code-based (e.g., McEliece), and hash-based (e.g., SPHINCS+). These algorithms will be tested in diverse environments, from resource-constrained devices to high-performance infrastructures such as cloud platforms. Key metrics such as encryption/decryption speed, key sizes, memory usage, and computational efficiency will be measured, focusing on the trade-offs between security, efficiency, and scalability. The results will be benchmarked against standardized metrics to provide a clear understanding of each algorithm's suitability for real-world deployment. The findings indicate that lattice-based algorithms like Kyber offer a strong balance between security and efficiency but require considerable computational resources. Code-based algorithms like McEliece are highly secure but come with large key sizes and slower speeds. Hash-based schemes like SPHINCS+ provide strong security but are computationally expensive, making them less suitable for resource-limited systems. The highlights the importance of optimizing PQC algorithms according to specific application requirements, where the choice of algorithm must balance security, efficiency, and resource constraints, with continuous optimization needed to address evolving real-world demands.

Keywords: Post Quantum Cryptographic algorithms

1. INTRODUCTION

The computational costs of post-quantum cryptographic (PQC) calculations vary essentially from conventional calculations like RSA and ECC due to their unmistakable numerical establishments [1]. For key eras, conventional calculations such as RSA require the era of expansive prime numbers, which is computationally direct, whereas ECC benefits from littler key sizes and effective elliptic bend operations, making it speedier. To differentiate, PQC calculations like lattice-based Kyber and Dilithium include complex network increases and inspecting from likelihood conveyances, driving to higher computational costs. Code-based calculations such as McEliece are especially costly for key eras due to their dependence on expansive lattices. In any case, hash-based calculations like SPHINCS+ are generally lightweight but depend intensely on hashing operations [2].

For encryption and decoding, RSA performs decently, with encryption being speedier than decoding due to the littler open example. ECC beats RSA in both operations because of its smaller key sizes and more productive computations. PQC calculations display shifting execution. Lattice-based calculations like Kyber frequently accomplish encryption and decoding speeds comparable to RSA but marginally slower than ECC. Code-based calculations tend to have quick encryption but slower decoding due to the complexity of disorder translating. Hash-based plans, whereas not regularly utilized for encryption, are computationally serious when performing marking operations [3][4].

In marking and confirmation, RSA is moderate for marking (a private key operation) but quick for confirmation (an open key operation). ECC is speedier in both angles due to its compact key sizes and proficient calculations. Among PQC calculations, lattice-based plans like Dilithium offer marking and confirmation speeds closer to ECC, whereas hash-based plans like SPHINCS+ are computationally overwhelming amid marking due to broad hash calculations. Multivariate quadratic plans, such as Rainbow, give effective marking but slower confirmation. Also, PQC calculations for the most part force critical memory and transmission capacity overhead, with key and ciphertext sizes extending from kilobytes to megabytes, compared to the compact keys of RSA and ECC. This makes PQC calculations more computationally serious by and large but essential for future-proofing cryptography against quantum dangers [5][6].

Post-quantum cryptographic (PQC) calculations by and large display higher computational costs, expanded overhead, and changed throughput compared to conventional RSA and ECC. Lattice-based calculations, like Kyber and Dilithium, offer a great adjustment of execution and security, with competitive execution times and higher throughput reasonable for secure communication and computerized marking. In addition, hash-based calculations, such as SPHINCS+, are computationally serious but exceed expectations in giving strong security, making them perfect for basic applications like firmware marking. Code-based calculations, such as McEliece, confront challenges due to their huge memory and transmission capacity prerequisites, restricting their utilization in resource-constrained situations. Whereas PQC calculations present overhead in key sizes and ciphertexts, their capacity to stand up to quantum assaults makes them fundamental for future-proofing cryptographic frameworks [2-6].

Post-quantum cryptography (PQC) is very important now because quantum computers can endanger old security methods used to protect information. Algorithms like RSA, ECC, and DH are used to keep communications safe. They depend on the difficulty of solving problems like breaking down numbers into their factors and finding discrete logarithms. But quantum algorithms, such as Shor's algorithm, can solve these problems quickly, making traditional security systems weak. This possible quantum threat puts the privacy and safety of important data at risk, especially valuable information like medical records, financial transactions, and government secrets.

PQC uses special math methods to protect data from being hacked by powerful quantum computers. These algorithms use math problems like lattice-based, hash-based, and code-based methods, which are thought to be difficult for even quantum computers to crack. Switching to PQC helps keep our communications safe and stops enemies from collecting encrypted information now to decode it later when they have powerful quantum computers. Since we don't know when practical quantum computers will be ready, using Post-Quantum Cryptography (PQC) is important to protect our digital systems and keep trust in cybersecurity as it changes [7].

Objective of the Study

The primary objective of this study is to analyze the performance measures of post-quantum cryptographic (PQC) algorithms in terms of computational cost, execution time, throughput, and overhead. Specifically, this paper aims to:

1. Compare the performance of various PQC algorithms, such as lattice-based, hash-based, and code-based schemes, against traditional cryptographic algorithms like RSA and ECC.

2. Evaluate the feasibility of PQC algorithms in real-world applications, including secure communications, digital signatures, and resource-constrained environments.

3. Identify trade-offs between security and performance to guide the selection of appropriate PQC algorithms for specific use cases.

4. Contribute insights into optimizing PQC implementations for practical deployment.

Significance of the Study

This consideration is noteworthy because it addresses the critical requirement for cryptographic arrangements versatile to quantum computing dangers. With headways in quantum innovation, conventional cryptographic frameworks confront out of date quality, gambling the secrecy and security of basic computerized foundation. By efficiently assessing the execution of PQC calculations, this inquiry gives significant bits of knowledge for organizations transitioning to quantum-resistant cryptography. The discoveries can direct designers, policymakers, and industry partners in selecting proficient and secure PQC plans, guaranteeing strong security for delicate information and keeping up belief in advanced communications. Besides, this consideration contributes to the broader exertion of standardizing PQC algorithms for far reaching selection within the post-quantum time.

2. OVERVIEW

Post-quantum cryptographic (PQC) algorithms are made to protect information from being broken by quantum computers. They depend on tough math problems that are thought to be safe from both regular and quantum computer attacks.

PQC algorithms are: 1. Lattice-Based Algorithms (like Kyber and Dilithium) work by using tough math problems, such as the Shortest Vector Problem (SVP). They are praised for being secure, efficient, and having smaller key sizes compared to other post-quantum cryptography methods. 2. Code-Based Algorithms (like McEliece) are built on the challenge of decoding random linear codes. They provide strong security but require large keys. This makes them good for situations where safety is more important than size. 3. Hash-Based Algorithms (like SPHINCS+): These algorithms use special math functions called hash functions to perform tasks. They offer very strong security but can be more expensive in terms of processing power. 4. Multivariate Quadratic (MQ) Systems (like Rainbow) use a set of multiple polynomial equations. They are good for creating digital signatures, but they can have problems when it comes to verifying those signatures. 5. Isogeny-Based Cryptography (like SIKE) relies on the difficulty of finding isogenies, which are special connections between elliptic curves. These methods use small keys but require a lot of computing power [8].

3. LITERATURE SURVEY

Numerous studies have explored the performance of Post-Quantum Cryptography (PQC) algorithms, focusing on benchmarking, computational requirements, and comparisons with classical cryptographic schemes. These investigations reveal critical insights into the efficiency and scalability of PQC algorithms under various conditions.

The literature on Post-Quantum Cryptography (PQC) performance analysis reveals a diverse range of studies focusing on different aspects of computational performance, cryptographic security, and system optimization. Existing research has primarily concentrated on optimizing computational codes, evaluating cryptographic algorithms, and assessing performance in high-performance computing environments. However, there are notable gaps in the literature, particularly concerning the integration of PQC with emerging technologies and the comprehensive evaluation of its performance across various platforms.

Studies have benchmarked PQC algorithms on standard PCs, assessing their performance across different metrics such as execution speed and memory utilization [9]. They evaluated traditional cryptographic algorithms like AES and RSA, but did not address Post-Quantum Cryptography (PQC) performance analysis. It highlighted the need for further exploration in PQC metrics, identified the gaps in existing literature on performance evaluation methodologies. The author evaluated the performance of various cryptographic algorithms, including AES, triple DES, Blowfish, RSA, MD5, and SHA, by measuring parameters such as the time required to encipher, the time required to decipher, and the memory used by each algorithm. A system for measuring the performance of these cryptographic algorithms is proposed and implemented, allowing for a comparative analysis to determine the most efficient algorithm in terms of resource usage. In [10] it identifies a lack of standardization in the optimization proposals for PQC algorithms, suggesting that future research should focus on coordination to facilitate an efficient and secure transition to post-quantum cryptography for IoT applications. For instance, the performance of Homomorphic Polynomial Public Key Cryptography (HPPK) was evaluated, demonstrating its efficiency in key generation and digital signature operations [11] focused on benchmarking the performance of Homomorphic Polynomial Public Key Cryptography (HPPK), highlighting its efficiency in key encapsulation and digital signatures. It identified gaps in existing PQC studies, emphasizing HPPK's compactness and superior performance metrics across various security levels. The researcher discussed global efforts in designing quantum-safe cryptography algorithms, highlighting their performance metrics such as CPU cycles, runtime memory, and key size. It identified gaps in existing literature regarding efficiency and feasibility, emphasizing the need for further exploration in these areas. It assessed various PQC digital signature algorithms, focusing on network performance, key robustness, and energy consumption. It highlights the MPPK/DS algorithm's advantages over RSA and SPHINCS, addressing gaps in existing literature regarding practical implementation and energy efficiency in resource-constrained networks. In

[12][13] focused on benchmarking the MPPK DS algorithm against NIST PQC schemes, highlighting its small key sizes and superior performance in key generation, signing, and verification. It identified gaps in existing studies regarding performance metrics for multivariate polynomial-based algorithms and also compared post-quantum signature algorithms' performance using execution time, communication costs, and implementation overheads. It highlighted gaps in existing literature regarding security strength metrics and proposes using depth-width cost for quantum circuits, advancing prior methodologies in performance analysis. In [14], it focused on comparing RSA and NTRU performance, highlighting NTRU's efficiency with smaller key sizes. It identified gaps in existing literature regarding performance metrics for post-quantum cryptographic algorithms, emphasizing the need for larger key sizes for security compared to NTRU in computational overhead and bandwidth efficiency. NTRU offers high speed with minimal computing power. In [15], it highlighted existing PQC studies, noting performance bottlenecks in key generation and signing processes. It introduced a hybrid approach combining extended Merkle Signature Scheme (XMSS) with Dilithium or Falcon, addressing gaps in role selection mechanisms and consensus optimization in blockchain-based federated learning systems. In [16], researcher reviewed existing PQC performance analyses, emphasizing the need for comprehensive security assessments and performance evaluations. It identifies gaps in standardization and integration challenges, highlighting the novelty of its guidelines for selecting and implementing quantum-resistant algorithms in real-world scenarios. Vulnerabilities of RSA and ECC to quantum attacks identified. In [17], consistent data is ensured by using a standardized set of messages for digital signatures. They focused on the most and least energy-consuming algorithms for both KEM and digital signatures. Lattice-based submissions demonstrated the lowest median energy consumption across security levels. SPHINCS+ and Picnic are notable for their energy consumption in signing and verification. Multivariate algorithms show higher energy consumption compared to lattice schemes. They could not expand energy analysis to lightweight cryptographic schemes and communication costs. Results are only categorized by energy efficiency for each security level and operation type. In [18], highlighted that clear-text denominations can hinder usability by increasing the number of required coins and the size of proofs needed for transactions. The authors highlight the ability for users to reconstruct their transaction history using a private view key, enhancing usability while maintaining privacy. It emphasized the need for robust security measures to safeguard user privacy in cryptocurrency systems. In [19], identified vectorization as a key optimization technique, resulting in significant performance improvements of 52% for the CRYSTALS-Kyber KEM SHA3 variant and 83% for the AES variant, specifically targeting polynomial multiplication and random number generation. In [20], a performance evaluation of PQC-integrated IKEv2 in terms of execution speed and packet size is presented, based on strongSwan, the most popular open-source IPsec implementation.

4. PERFORMANCE METRICS

The PQC algorithms and metrics are essential to evaluate their suitability for secure communication, digital signatures, and other cryptographic needs in the post-quantum era. Computational Cost which is the number of operations (e.g., matrix multiplications, hash evaluations) required for key generation, encryption/decryption, and signing/verification. Execution Time, the time taken to perform cryptographic operations, which impacts real-time and high-throughput applications. Throughput, the number of operations completed per second, relevant for high-performance systems. Overhead which includes memory usage (e.g., key sizes, ciphertext sizes, signature sizes) and bandwidth requirements, which affect scalability and resource-constrained environments [21][22].

We choose different post-quantum cryptographic (PQC) algorithms from NIST for thorough study. Each candidate has its own strengths in security and performance. Kyber is a type of key exchange method that uses a special math structure called a lattice. It is known for being fast, having small keys, and working well for secure communication. Dilithium is a type of technology that uses a special grid-like structure to create secure digital signatures while using a reasonable amount of memory. This makes it a great choice for applications that need to sign documents securely. NTRU is a type of encryption that uses a lattice method. It allows for quick encrypting and decrypting, which makes it a good option for public-key encryption and sharing keys. McEliece is a type of encryption that uses codes to keep information safe. It is very secure, but the large keys it requires make it hard to use in places with limited data capacity. SPHINCS+ is a method for creating digital signatures that is very secure but needs a lot of computer power and memory to sign messages. Rainbow is a system that uses complex equations. It is great at quickly signing documents but takes longer to check those signatures. It provides a good mix of speed and safety. In short, SIKE is a way to exchange keys that uses isogenies. It has small keys and doesn't take up much space, but it is slower than methods based on lattices. This makes it great for places where bandwidth is limited. These algorithms give a clear idea of the advantages and disadvantages in terms of cost, time taken to run, amount of data processed, and extra resources needed for post-quantum cryptography [23].

5. PERFORMANCE METRICS FOR POST QUANTUM CRYPTOGRAPHIC ALGORITHMS

To evaluate post-quantum cryptographic (PQC) algorithms, several key performance metrics must be considered. These metrics help in understanding the feasibility and practicality of implementing these algorithms in real-world systems, especially when compared to traditional cryptographic schemes like RSA and ECC [24]. The main performance metrics are:

1. Computational cost: This refers to the number of CPU cycles or basic operations required to perform key cryptographic functions such as key generation, encryption, decryption, and signing. A higher computational cost typically translates to more processing power needed, affecting the speed and efficiency of the algorithm. Algorithms with lower computational costs are more suited for devices with limited processing capabilities, such as embedded systems, IoT devices, and mobile platforms. For example, **Kyber** and **Dilithium** are known for their relatively low computational costs in terms of both key generation and encryption, making them more efficient in terms of CPU usage.

2. Execution Time: This metric measures the time required for specific cryptographic operations, including key generation, encryption, decryption, and signature generation/verification. It is a crucial metric for determining how long it takes to perform cryptographic operations in a real-time system or a high-throughput application. Shorter execution times are essential for real-time communication or applications that require fast cryptographic operations, such as secure messaging or online transactions. **Kyber** and **NTRU** offer relatively fast encryption and key generation times, whereas **SPHINCS+** has longer execution times due to its computationally intensive signing process.

3. Memory Usage: Memory usage refers to the amount of RAM and storage required to implement and execute a PQC algorithm. This includes the memory needed for storing keys, intermediate computations, and ciphertexts. Algorithms with high memory usage can be impractical for resource-constrained environments. **McEliece** is known for its large key sizes and thus requires considerable storage space, making it less suitable for environments with limited memory or bandwidth. On the other hand, **Dilithium** and **Kyber** are more memory-efficient, balancing security and memory requirements more effectively.

4. Bandwidth/ computational Overhead: This metric refers to the size of keys, ciphertexts, and signatures that need to be transmitted during communication. Larger key sizes and longer ciphertexts increase bandwidth consumption, which can be a significant issue in networks with limited bandwidth or when transmitting large volumes of data. Algorithms with larger key and ciphertext sizes, such as **McEliece**, can impose high communication overheads, making them less suitable for applications where bandwidth is limited. **Kyber** and **Dilithium**, with relatively smaller key sizes and ciphertexts, are better suited for environments where bandwidth is a concern.

5. Throughput: Throughput refers to the number of cryptographic operations that can be completed per second. It is an important metric for high-performance systems, such as data centers or cloud services, where large volumes of cryptographic operations are required. High throughput is crucial for applications requiring high-performance encryption, key exchange, or signing operations. **NTRU** and **Kyber** generally provide higher throughput due to their more efficient algorithms compared to other PQC schemes like **SPHINCS+**, which has lower throughput due to the complexity of its signature generation [25][26][27].

6. PERFORMANCE EVALUATION OF HASHING ALGORITHMS

Cryptographic operations play a crucial role in ensuring the security and integrity of data across various applications, from internet communication to secure storage. Hashing is one of the most fundamental cryptographic processes, which involves generating a fixed-length hash output from inputs of varying sizes. The performance of hashing can be evaluated based on the time taken to hash data of different sizes, such as very short packets, typical-size internet packets, and long messages. Notably, the length of the hash output remains constant, irrespective of the input size, which is a key property of cryptographic hash functions like SHA-256 and SHA-512.

Symmetric encryption is another vital operation, where the same secret key is used for both encryption and decryption. Performance is measured by the time taken to encrypt data of varying sizes, from very short packets to large messages, using a secret key and a nonce. The length of the secret key (e.g., 128, 192, or 256 bits) and the nonce (e.g., 96 or 128 bits) are critical parameters influencing both security and computational efficiency. Additionally, authenticated encryption combines confidentiality and integrity by securing the data and verifying its authenticity. The time required for authenticated encryption of short packets, typical-size internet packets, and long messages is a key performance metric, ensuring secure and efficient data handling.

Public-key cryptography introduces asymmetric operations, involving a pair of private and public keys. The time to generate these key pairs, along with the lengths of the private and public keys, are important considerations. Similarly, the time to compute a shared secret using a private key and another user's public key, along with the length of the derived shared secret, reflects the performance of key agreement protocols like Diffie-Hellman and Elliptic Curve Diffie-Hellman (ECDH). Public-key operations also include encryption and decryption, where the time to encrypt a message using a public key and decrypt it with a private key, as well as the length of the encrypted message, are evaluated to assess efficiency.

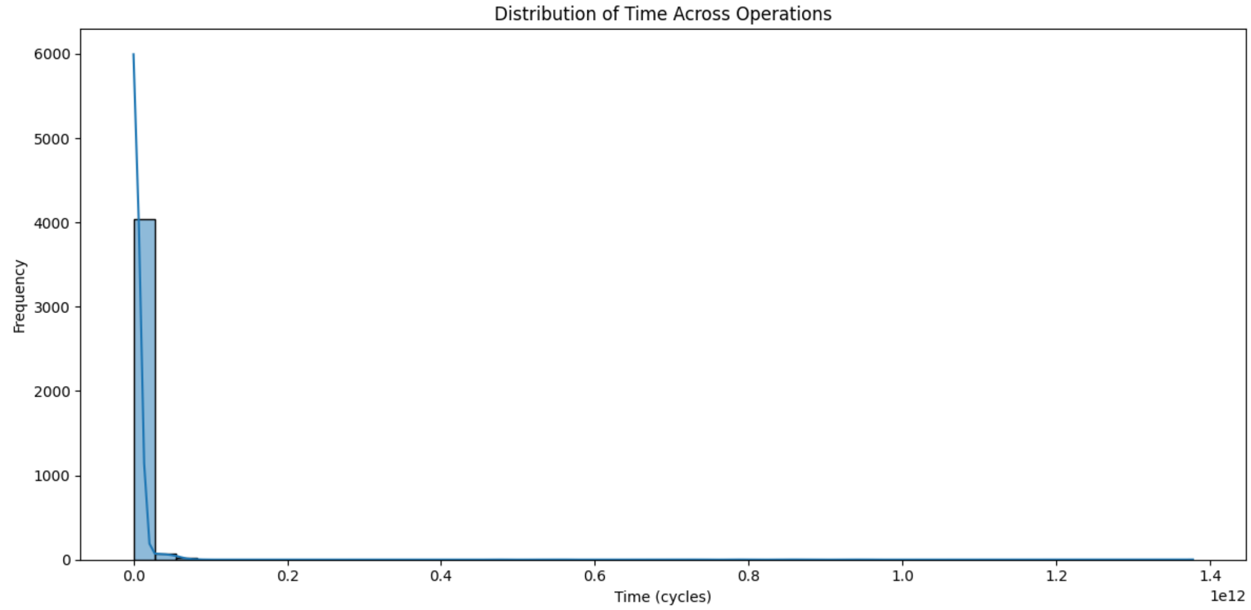
Digital signatures are crucial for ensuring the authenticity and integrity of data. The time to sign a message using a private key, the length of the signed message, and the time to verify the signature using a public key are significant performance indicators. These operations are particularly important in applications like secure messaging, blockchain transactions, and certificate-based authentication. By systematically evaluating the time and resource requirements for these cryptographic operations, researchers and practitioners can optimize the balance between security and performance in various computational and networking environments [28].

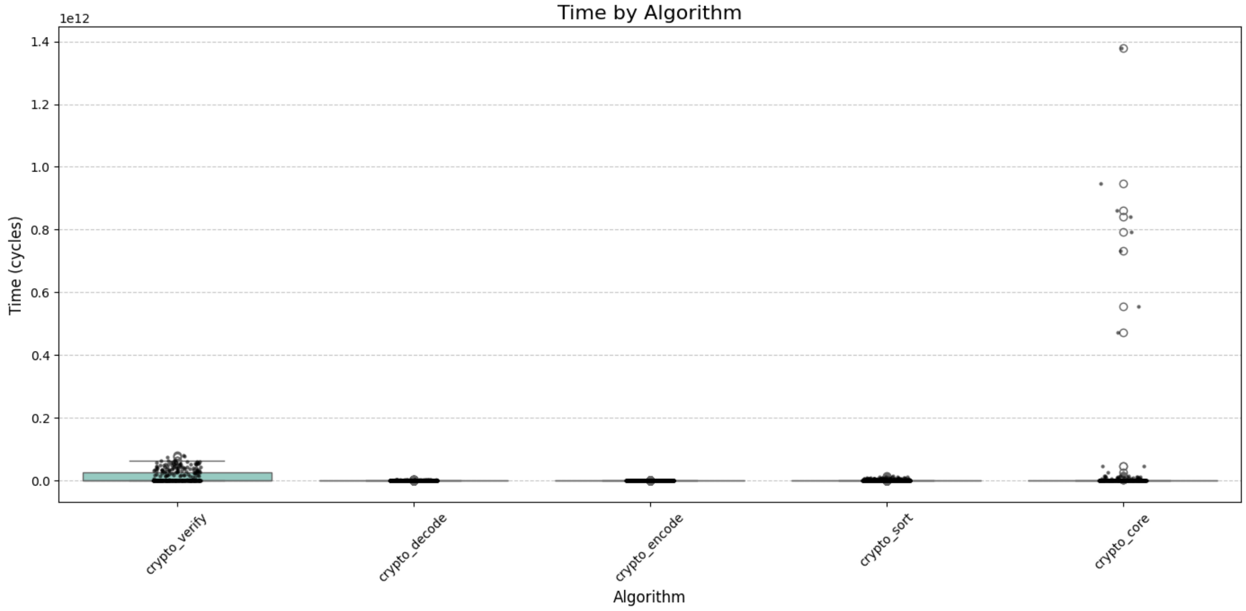
7. BENCHMARKING SETUP AND ENVIRONMENT

For cryptographic performance evaluation, we utilize **SUPERCOP** (System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives), a benchmarking tool that is specifically designed to run on Linux-based systems. As SUPERCOP is not natively compatible with Windows, we recommend using **Windows Subsystem for Linux (WSL)** as a viable alternative. WSL allows Windows users to run a Linux environment directly on their system, offering a seamless experience for executing SUPERCOP and related cryptographic tests. This approach is preferred over the older **Cygwin** tool, which may present compatibility issues and performance drawbacks. To ensure that the necessary packages are installed for running SUPERCOP on a Linux-based system (such as Ubuntu), the following commands should be executed in the terminal. These commands install essential development tools, including **GCC**, **OpenSSL**, **Make**, and **Python3**, which are required to compile cryptographic algorithms and manage benchmarking tasks. These packages are fundamental for preparing the system environment, enabling successful compilation of cryptographic primitives and ensuring that performance metrics can be accurately measured during benchmarking.

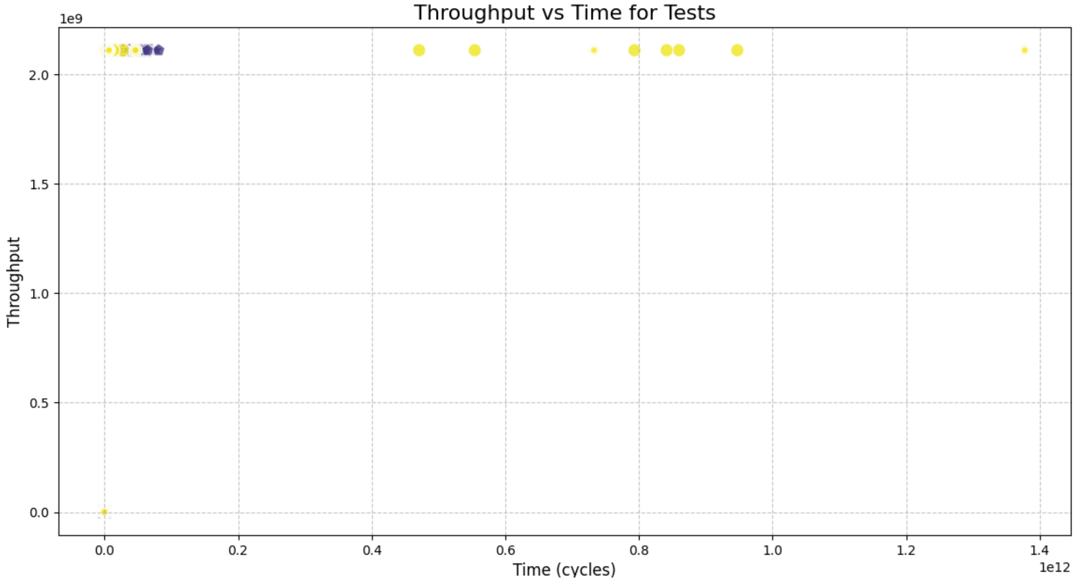
8. Results

The following diagram having each box represents the distribution of time for an algorithm.The filled dots (strip plot) show individual data points, providing more granularity.Outliers, if any, are visible as points beyond the whiskers of the box plot.Patterns can reveal which algorithms are consistently faster or have variable performance.





The histogram shows the frequency of time values, with a smooth KDE (Kernel Density Estimation) curve overlayed for better visualization.Peaks in the graph represent common time ranges.The spread of the histogram shows the variability in operation time, helping identify whether times are tightly clustered or widely spread.



The data set used in this study encompasses all classical and quantum-resistant algorithms prescribed within the NIST standards, as well as their final variant forms. This comprehensive collection allows for an in-depth comparison across a wide range of cryptographic primitives.

The benchmarking process itself is extensive, running over a period of approximately 96 hours. The total volume of collected data amounts to 1.2 gigabytes, or more precisely, 1,304,928,041 kilobytes. This large data set provides a robust foundation for analyzing the performance of the algorithms under test, ensuring that the results are statistically significant and reflective of real-world usage.

CONCLUSION:

In conclusion, this study provides a comprehensive evaluation of the performance of various NIST-recommended post-quantum cryptographic (PQC) algorithms, highlighting the trade-offs between security, computational cost, execution time, memory usage, throughput, and bandwidth overhead. While PQC algorithms offer robust security against quantum threats, their higher computational costs and resource demands present challenges for practical deployment in real-world systems, especially in resource-constrained environments. The benchmarking process, based on an extensive 96-hour testing period and a large dataset, reveals distinct patterns in algorithm performance, with some algorithms like Kyber and Dilithium offering better efficiency in terms of computational cost and memory usage, while others like SPHINCS+ require more resources. Visualizing the data through box plots, histograms, and Kernel Density Estimation (KDE) further aids in understanding the efficiency of each algorithm across different use cases. The study emphasizes the importance of continued research to standardize performance metrics, optimize the integration of PQC algorithms into emerging technologies, and address existing gaps in their scalability across diverse platforms. As quantum computing advances, the findings of this study will play a crucial role in guiding the selection and implementation of PQC algorithms for future cryptographic systems.

REFERENCES:

[1] D. J. Bernstein, C.-K. Wu, and T. Lange, "The security and performance of post-quantum cryptographic algorithms," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1-34, 2017, doi: 10.1145/3124429.

[2] D. D. B. and G. K. and C. K. and L. T., "SPHINCS+: Practical Stateless Hash-based Signatures," *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018, pp. 43-58.

[3] L. Ducas, D. Micciancio, and C. Peikert, "Lattice-based Cryptography," *Springer Handbook of Cryptography*, pp. 543-564, 2020, doi: 10.1007/978-3-030-38945-9_30.

[4] S. G. S. and L. F. and M. M. and H. R. S., "Review of Cryptographic Hash Functions for Hash-Based Signatures," *IEEE Access*, vol. 10, pp. 23634-23652, 2022, doi: 10.1109/ACCESS.2022.3163295.

[5] T. L., "Post-Quantum Cryptography: Challenges and Opportunities," *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2423-2440, April 2019, doi: 10.1109/TIT.2018.2883216.

[6] M. A. G. and T. D. L., "Post-Quantum Cryptography: A Survey of Techniques and Current Standards," *IEEE Access*, vol. 8, pp. 90422-90449, 2020, doi: 10.1109/ACCESS.2020.2993324.

[7] D. J. Bernstein, T. Lange, and C. M. V. D. Put, "The Security and Performance of Post-Quantum Cryptographic Algorithms," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1-34, 2017, doi: 10.1145/3124429.

[8] R. Steinfeld, Y. Sakurai, and H. Shacham, "Post-quantum cryptography and quantum computer attacks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, New York, NY, USA, 2009, pp. 1–10.

[9] A. P. Parkar, M. N. Gedam, N. Ansari, and S. Therese, "Performance Level Evaluation of Cryptographic Algorithms," in *Intelligent Computing and Networking*, Conference Paper, Oct. 2023.

[10] T. Liu, G. Ramachandran, and R. Jurdak, "Post-Quantum Cryptography for Internet of Things: A Survey on Performance and Optimization," *arXiv*, vol. abs/2401.17538, Jan. 2024.

[11] R. Kuang, M. Perepechaenko, D. Lou, and B. Tank, "Benchmark Performance of Homomorphic Polynomial Public Key Cryptography for Key Encapsulation and Digital Signature Schemes," *IACR Cryptology ePrint Archive*, Jan. 2024.

[12] T. Penduff, "Post-Quantum Cryptography Algorithms Standardization and Performance Analysis," Apr. 2022.

[13] R. Kuang, M. Perepechaenko, R. Toth, and M. Barbeau, "Benchmark Performance of a New Quantum-Safe Multivariate Polynomial Digital Signature Algorithm," pp. 454-464, Sept. 2022.

[14] M. Raavi, S. Wuthier, P. Chandramouli, Y. Balytskyi, X. Zhou, and S.-Y. Chang, "Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms," June 2021.

[15] N. Challa and J. Pradhan, "Performance Analysis of Public Key Cryptographic Systems RSA and NTRU," Jan. 2007.

[16] D. Gurung, S. R. Pokhrel, and G. Li, "Performance Analysis and Evaluation of Post Quantum Secure Blockchained Federated Learning," June 2023.

[17] A. K. Pandey, A. Banati, B. Rajendran, S. D. Sudarsan, and K. K. S. Pandian, "Cryptographic Challenges and Security in Post Quantum Cryptography Migration: A Prospective Approach," Sept. 2023.

[18] C. A. Roma, C.-E. A. Tai, and M. A. Hasan, "Energy Efficiency Analysis of Post-Quantum Cryptographic Algorithms," *IEEE*, Jan. 2021.

[19] C. Paquin, D. Stebila, and G. Tamvada, "Benchmarking Post-quantum Cryptography in TLS," *IACR Cryptology ePrint Archive*, Apr. 2020.

[20] S. Koteshwara, M. Kumar, and P. Pattnaik, "Performance Optimization of Lattice Post-Quantum Cryptographic Algorithms on Many-Core Processors," *IEEE*, pp. 223-225, Aug. 1, 2020.

[21] J. Yao, A. Hlayhel, and K. Matusiewicz, "Post Quantum KEM authentication in SPDM for secure session establishment," *IEEE Journal of Selected Topics in Signal Processing*, doi: 10.1109/MDAT.2023.3292998, Jul. 7, 2023.

[22] M. Kumar, "Post-Quantum Cryptography Algorithm's Standardization and Performance Analysis," *Computer Science Cryptography and Security*, arXiv:2204.02571, Apr. 6, 2022.

[23] S. Jia, "Comparison of Performances for Quantum and Conventional Algorithms: Shor's Algorithm and Boson Sampling," in *Proc. 2022 Int. Conf. Theoretical, Physics Computers and Electronic Engineering (TPCEE 2022)*, vol. 38, Mar. 16, 2023.

[24] J. J. Tom, N. P. Anebo, B. A. Onyekwelu, A. Wilfred, and R. E. Eyo, "Quantum Computers and Algorithms: A Threat to Classical Cryptographic Systems," *Int. J. Eng. Adv. Technol.*, vol. 12, no. 5, pp. 25-38, Jun. 30, 2023.

[25] L. B. S. and N. Kaulgud, "A review on analysis of transport layer security in open quantum safe cryptographic algorithm," in *Proc. 2023 Int. Conf. Recent Trends in Electronics and Communication (ICRTEC)*, May 3, 2023.

[26] M. Kumar, "Post-quantum cryptography algorithm's standardization and performance analysis," *Array*, vol. 15, 100242, Elsevier, 2022.

[27] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–11, 2014.

[28] G. Bebrov, "On the (relation between) efficiency and secret key rate of QKD," *Sci. Rep.*, Feb. 13, 2024.