Asia-Pacific Journal of Information Technology and Multimedia 15(1) 2026: 84 – 98
https://doi.org/10.17576/apjitm-2026-1501-05

Feasibility of Post-Quantum Cryptography in Digital Signature Systems: From Theory to Proof of Concept

Kebolehlaksanaan Algoritma Kriptografi Pasca-Kuantum dalam Sistem Tandatangan Digital: Dari Teori ke Bukti Konsep

Nurul Syafiqah Norihsan[1], Azana Hafizah Mohd Aman[1], Fakrul Radzi Ab Rahim[2], Hazhar Ismail[2]*

[1]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor 43600, Malaysia

[2]MSC Trustgate, Suite 2-9, 4801, Jln Perdana, Cyberjaya Business District Perdana 1, 63000 Cyberjaya, Malaysia

**Corresponding author: Nurul Syafiqah Norihsan (syafiqahihsan18@gmail.com)*

Received 31 July 2025
Accepted 28 April 2026, Available online 30 June 2026

ABSTRACT

The advancement of quantum technology is increasingly pressuring the security of traditional cryptographic algorithms such as RSA and ECDSA, which are widely used in Public Key Infrastructure (PKI). As a preventive measure, the digital security community is paying closer attention to post-quantum cryptographic (PQC) algorithms that are resilient against quantum-computing threats. Among the PQC algorithms approved by NIST are CRYSTALS-Dilithium and SPHINCS+, as announced during the third round of the NIST PQC selection process in 2022. Therefore, this project aims to develop a proof-of-concept (PoC) platform as a web application using Java Spring Boot, the Bouncy Castle cryptographic library, and Bootstrap. The system supports RSA, ECDSA, CRYSTALS-Dilithium, and SPHINCS+ algorithms for key pair generation, file signing, and digital signature verification. The user interface allows users to select algorithms, sign files, and quickly check the signature status. A benchmarking module is also provided to measure key generation time, signing and verification time, and signature size for each output produced by the tested algorithms. Benchmarking results demonstrate that CRYSTALS-Dilithium offers superior computational efficiency compared to legacy algorithms, achieving verification times as low as 1 ms, while SPHINCS+ presents a distinct trade-off, characterized by high signing latency and a significantly larger signature footprint of 17,088 bytes. The paper's results show that integrating PQC algorithms into digital certificate systems is technically feasible and has the potential to enhance long-term security in the post-quantum era.

Keywords: RSA, ECDSA, CRYSTALS-Dilithium, SPHINCS+, PQC

ABSTRAK

Kemajuan teknologi kuantum semakin memberi tekanan terhadap keselamatan algoritma kriptografi tradisional seperti RSA dan ECDSA yang digunakan secara meluas dalam infrastruktur kekunci awam (PKI). Sebagai langkah pencegahan, komuniti keselamatan digital memberi perhatian lebih kepada algoritma kriptografi pasca-kuantum (PQC) yang mampu bertahan daripada ancaman komputer kuantum. Antara algoritma PQC yang telah diluluskan oleh NIST ialah CRYSTALS-Dilithium dan SPHINCS+, seperti yang diumumkan dalam pusingan ketiga dalam pemilihan algoritma PQC oleh NIST pada 2022. Oleh itu, projek ini bertujuan untuk membangunkan satu platform bukti konsep (PoC) dalam bentuk aplikasi web menggunakan Java Spring Boot, pustaka Bouncy Castle dan Bootstrap. Sistem ini menyokong algoritma RSA, ECDSA, CRYSTALS-Dilithium, dan SPHINCS+ untuk penjanaan pasangan kunci, penandatanganan fail, dan pengesahan tandatangan digital. Antara muka sistem membolehkan pengguna memilih algoritma, menandatangani fail, dan menyemak status tandatangan dengan pantas. Pengujian tanda aras juga turut disediakan bagi mengukur masa penjanaan pasangan kunci, masa penjanaan dan pengesahan tandatangan dan saiz tandatangan bagi setiap output daripada algoritma yang diuji. Keputusan penanda aras menunjukkan bahawa CRYSTALS-Dilithium menawarkan kecekapan pengiraan yang unggul berbanding algoritma warisan, mencapai masa pengesahan serendah 1 ms, manakala SPHINCS+ memberikan penjanaan berbeza yang dicirikan oleh latensi tandatangan yang tinggi dan jejak tandatangan yang jauh lebih besar sebanyak 17,088 byte. Hasil kajian menunjukkan bahawa integrasi algoritma PQC dalam sistem sijil digital adalah tidak mustahil untuk dilaksanakan secara teknikal dan mempunyai potensi untuk meningkatkan keselamatan jangka panjang dalam era pasca-kuantum.

Kata kunci: RSA, ECDSA, CRYSTALS-Dilithium, SPHINCS+, PQC

INTRODUCTION

In the current digital ecosystem, cryptographic algorithms play a crucial role in ensuring the security and authenticity of data, particularly within Public Key Infrastructure (PKI). However, with the rapid advancement of quantum computing technology, traditional algorithms such as RSA and ECDSA are becoming increasingly vulnerable. Quantum algorithms like Shor's and Grover's can break the hard mathematical assumptions behind these classical systems, rendering them insecure in a post-quantum world (Academy, 2024).

Recognizing this threat, the cryptographic community has actively developed post-quantum cryptographic (PQC) algorithms. The National Institute of Standards and Technology (NIST), through its Post-Quantum Cryptography Standardization Project, selected CRYSTALS-Dilithium and SPHINCS+ as standard digital signature schemes in 2022 (NIST, 2022). These algorithms provide quantum-resistant alternatives through lattice-based and hash-based cryptographic constructs, respectively.

This paper presents the design and development of a web-based proof-of-concept (PoC) digital signature system that integrates both classical (RSA, ECDSA) and post-quantum (Dilithium3, SPHINCS+) algorithms. The system was built using Java Spring Boot and the Bouncy Castle library, with a responsive frontend developed using Bootstrap. It allows users to generate key pairs, sign and verify files, and perform benchmarking to assess the performance of different cryptographic algorithms. The study aims to evaluate the feasibility of incorporating PQC

algorithms into modern digital signature systems and highlight their performance differences compared to traditional methods.

This paper makes the following contributions to the body of knowledge on the PQC Proof-of-Concept (PoC):

1. An integrated framework design and implementation that facilitates the transition from legacy cryptographic standards (RSA/ECDSA) to PQC. Unlike static implementations, this paper provides a "crypto-agile" bridge that allows systems to maintain backward compatibility while adopting quantum-resistant algorithms such as CRYSTALS-Dilithium and SPHINCS+.
2. An analysis of empirical PQC algorithms across multiple performance dimensions, including Key Generation, Signature Creation, and Verification Latency. By evaluating these metrics in bytes and milliseconds, this paper offers practical data for engineers to select algorithms based on specific constraints (e.g., choosing Dilithium for low latency or SPHINCS+ for high-security archival needs).
3. This paper demonstrates how the Model-View-Controller (MVC) architectural pattern can be effectively utilized to decouple cryptographic logic from the user interface. This separation ensures that security protocols can be updated or replaced (e.g., swapping a lattice-based scheme for a hash-based one) with minimal impact on the overall system architecture, promoting long-term system maintainability.
4. This paper provides the translation of theory to a PoC, the transition from theoretical literature review (related work) findings to a functional web-based toolkit. This paper serves as a demonstrative platform for researchers and developers to visualize PQC operations, bridging the gap between abstract cryptographic specifications and actual software deployment.

RELATED WORK

Numerous research studies have explored the implementation and evaluation of post-quantum cryptographic (PQC) algorithms, particularly in digital signature systems. These studies aim to assess the feasibility, efficiency, and practicality of PQC integration, especially under the imminent threat posed by quantum computing. Among the most notable algorithms studied are CRYSTALS-Dilithium and SPHINCS+, both of which were selected by NIST for standardization due to their strong security and structural resilience against quantum attacks.

In a study by Hülsing et al. (2022), the performance of SPHINCS+ was evaluated in various cryptographic environments. The researchers emphasized the trade-off between high security and large signature sizes. While SPHINCS+ produced signatures exceeding 7 KB, it still delivered robust hash-based stateless security. The study highlighted the algorithm’s suitability in long-term archival systems or blockchain environments where signature size is not a critical constraint. The main limitation is its relatively slower signing and verification time compared to lattice-based schemes. On the other hand, the operational suitability of the SPHINCS+, given its larger signature sizes but robust security foundation (hash-based), makes it recommended for high-security archival signing and firmware updates where verification speed is prioritized over bandwidth, or where long-term security is more critical than communication overhead (Wu et al., 2025; Wang et al., 2025).

Bos et al. (2018) focused on CRYSTALS-Dilithium, which is based on lattice problems using the Module-LWE and Module-SIS assumptions. Their implementation demonstrated high signing speed and smaller signature sizes (~2.7 KB), making it well-suited for embedded

devices and Internet of Things (IoT) applications. The researchers developed a reference implementation in C and showed that Dilithium offers better performance than other lattice-based contenders such as Falcon. However, they also noted that Dilithium's private key size is relatively large (~3 KB–4 KB), which might affect systems with limited storage.
A comparative benchmark study by Bindel et al. (2019) investigated the integration of RSA, Dilithium, and SPHINCS+ in a TLS handshake simulation. The experiment measured key generation time, signing time, verification time, and cryptographic object sizes using OpenSSL and Bouncy Castle. It was found that, RSA-2048 had the smallest signature size (256 bytes) and fastest signing time, but the slowest key generation. Dilithium3 achieved a balance between speed and signature size, with key generation taking ~10 ms and verification ~1.8 ms. While SPHINCS+ required the longest signing time (~18 ms) and produced the largest signature size (~7.8 KB) but had excellent public key compactness (32 bytes). A summary of each literature is shown in Table 1.

TABLE 1. Summary of Literature Review on PQC Algorithm Perfomance

Ref.	Algorithm	Method/Parameters	Result	Advantage	Limitation
Hülsing et al. ,2022; Wu et al., 2025; Wang et al. 2025	SPHINCS+	The algorithm is hash-based and stateless, benchmarked on a 64-bit Linux platform to evaluate its performance and feasibility in real-world scenarios.	The results show a relatively large signature size and slower signing time compared to other post-quantum algorithms.	Offers very high post-quantum security and avoids key state management, reducing implementation complexity and risk.	Produces large signature sizes and experiences slower signing operations, which may limit use in constrained environments.
Bos et al. ,2018; Astarloa et al.,2025; Carril et al., 2025; Shin et al., 2025	CRYSTALS-Dilithium	The lattice-based algorithm was implemented in C and tested in embedded system environments to assess its efficiency for low-resource devices.	Generates compact signatures with fast signing times and good overall performance on constrained hardware like IoT devices.	Well-suited for lightweight and embedded applications due to its balance of security and performance.	Suffers from relatively large private key sizes, which could be a challenge for devices with limited memory capacity.
(Bindel et al. ,2019)	RSA vs. Dilithium vs. SPHINCS+	The study used a TLS simulation environment with Bouncy Castle and OpenSSL libraries.	RSA had the fastest signing time, Dilithium performed best for verification, while SPHINCS+ was the slowest in both.	Provided a realistic comparison within a widely used security protocol and showed integration viability.	RSA is not secure in the post-quantum era, and SPHINCS+ exhibited high latency and storage costs.

To evaluate the feasibility of integrating post-quantum cryptographic algorithms into practical digital signature systems, a Proof of Concept web platform was developed using Java Spring Boot and integrated with Chart.js for dynamic data visualization. The system allows benchmarking of four digital signature algorithms: RSA, ECDSA, CRYSTALS-Dilithium, and SPHINCS+. The Proof of Concept system developed in this study visualizes three key metrics:

5. Key Generation Time (ms)
6. Signature Generation Time (ms)
7. Signature Verification Time (ms)

A comparative performance of RSA, ECDSA, CYRSTALS-Dilithium and SPHINCS+ is shown in Figure 1.

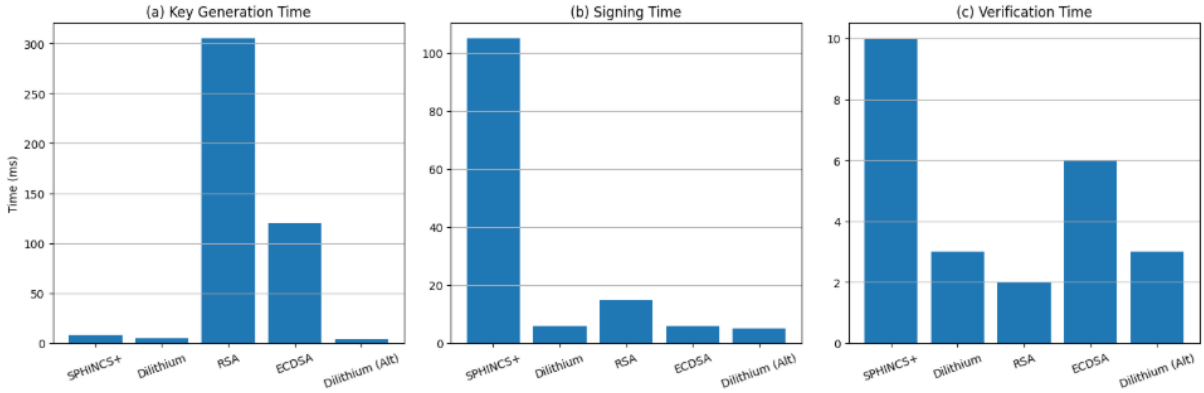


FIGURE 1. Comparative Performance of RSA, ECDSA, CRYSTALS-Dilithium, and SPHINCS+

From the figure, it is evident that RSA demonstrates the highest key generation time, exceeding 300 ms, followed by ECDSA at 118 ms. In contrast, CRYSTALS-Dilithium achieves exceptional performance with key generation times as low as 2–3 ms, while SPHINCS+ ranges around 7 ms. In terms of signature generation, SPHINCS+ exhibits significant latency (106 ms) compared to the other three algorithms which range between 2–14 ms. Similarly, signature verification for SPHINCS+ is slower (10 ms) than CRYSTALS-Dilithium (1–2 ms) and ECDSA (5 ms). The performance trend indicates that CRYSTALS-Dilithium is highly efficient across all metrics, making it a strong candidate for future PQC deployments. On the other hand, SPHINCS+ trades off performance for a stateless and hash-based architecture that delivers stronger long-term resistance but incurs greater computational and storage cost. To complement the time-based evaluation, the system also benchmarks the output size of each algorithm, including key sizes and signature sizes. These are essential considerations for memory-constrained environments and for maintaining compact certificates in digital identity infrastructures. Key and signature size metrics for tested algorithms in this system is shown in Table 2.

TABLE 2. Key and Signature Size Metrics for Tested Algorithms

Algorithm	Key Gen Time (ms)	Sign Time (ms)	Verify Time (ms)	Public Key Size (bytes)	Private Key Size (bytes)	Signature Size (bytes)
RSA	307	14	0	294	1217	256
ECDSA	118	4	5	91	67	70
CRYSTALS-Dilithium	2-3	4	1-2	1976	6019	3309
SPHINCS+	7	106	10	47	118	17088

From the table above, RSA and ECDSA maintain very small signature sizes (256 bytes and 70 bytes, respectively), which makes them suitable for resource-limited applications. However, their vulnerability to quantum attacks makes them unsuitable for future-proof implementations.

Conversely, Dilithium strikes a good balance between performance and size, while SPHINCS+ exhibits extremely large signature sizes (17,088 bytes), which may pose storage and bandwidth challenges in large-scale deployments.

METHODOLOGY AND IMPLEMENTATION

This study outlines the methodology used for developing and testing a web-based digital signature verification system that incorporates both classical and post-quantum cryptographic algorithms, namely RSA, ECDSA, CRYSTALS-Dilithium, and SPHINCS+. The system was built as a Proof-of-Concept platform to allow users to generate key pairs, sign files, and verify digital signatures interactively via a user-friendly web interface. The methodology focused on cryptographic functionalities such as key generation, signing, and verification, as well as performance evaluation metrics including processing time and key/signature size. The development was conducted using Java Spring Boot for backend logic, Bouncy Castle as the cryptographic library, and HTML, JavaScript, and Bootstrap for the frontend. The system was designed as a standalone simulation platform to assess the practicality and effectiveness of PQC algorithms.

Requirement analysis was conducted to ensure the system meets project objectives and functions reliably under real-world conditions. This involved identifying functional requirements (e.g., algorithm selection, key generation, file signing, and signature verification) and non-functional requirements (e.g., system performance, data integrity, usability, and system stability). The functional requirements specified that the system must support algorithm selection (RSA, ECDSA, CRYSTALS-Dilithium, and SPHINCS+), allow generation of public-private key pairs, perform signing of documents or messages, and verify signatures using the selected algorithm. The system should also display benchmarking results to enable comparative performance evaluation. The non-functional requirements emphasized performance metrics such as key generation speed, signing and verification time, data integrity, secure key handling, and a user-friendly interface that allows long-term use without errors. The RSA process is shown in Figure 2.

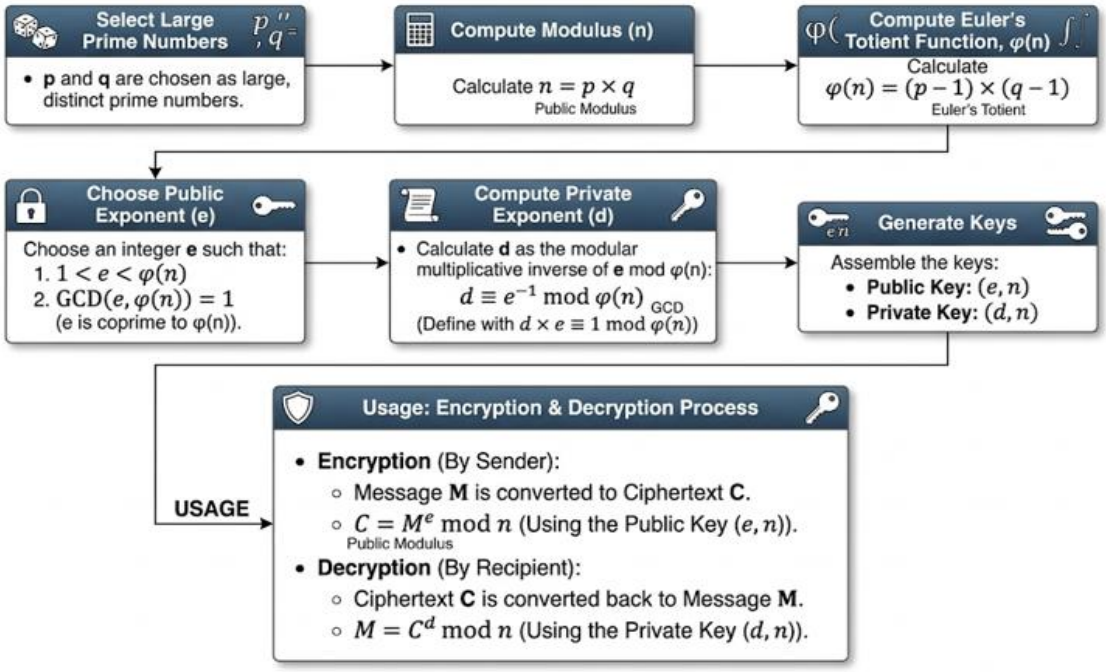


FIGURE 2. Process for RSA

The proposed system model comprises five main modules namely, algorithm selection, key generation, digital signature generation, signature verification, and performance benchmarking (shown in Figure 3 and Figure 4). This modular approach ensured systematic development aligned with the research objectives. The following are the processes in the model.

1. The algorithm selection module enables users to choose RSA, ECDSA, Dilithium3, or SPHINCS+.
2. The key generation module generates public-private key pairs according to the selected algorithm. RSA relies on integer factorization, Dilithium3 uses lattice-based cryptography (MLWE), and SPHINCS+ employs hash-based tree structures.
3. The signature generation module signs messages using private keys in a secure manner.
4. The signature verification module uses public keys to ensure the authenticity and integrity of the signed data.
5. The benchmark module records and displays metrics such as key generation time, signing time, verification time, key size, and signature size using tables and charts.

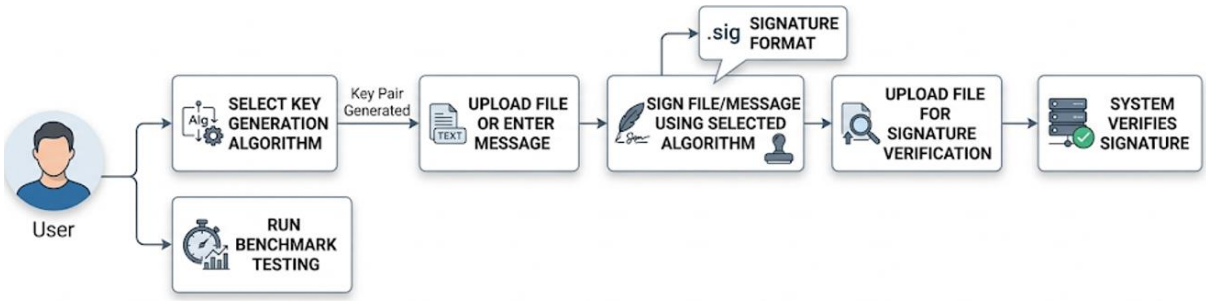


FIGURE 3. System Model

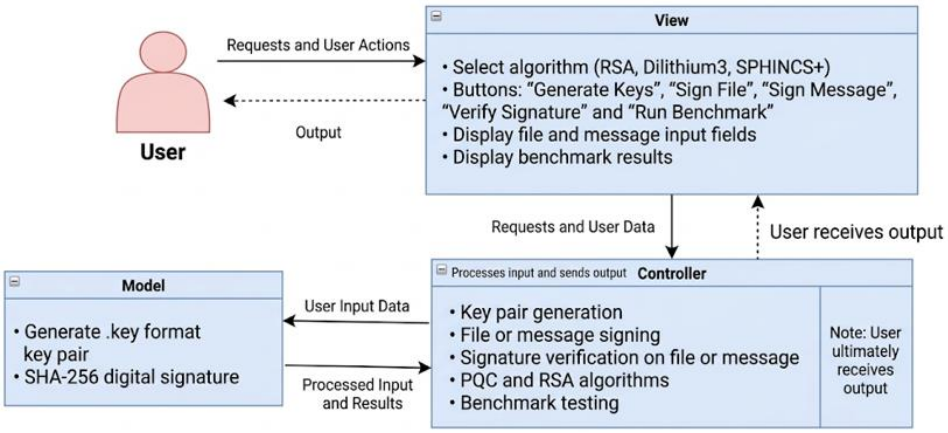


FIGURE 4. Model-View-Controller of the system

The interface design process involves creating a prototype that prioritizes usability and user convenience. A well-designed interface should be user-friendly, easy to understand, structured, and consistent to ensure it meets users' needs effectively and is simple to operate. The proposed system interface is shown in Figure 5.

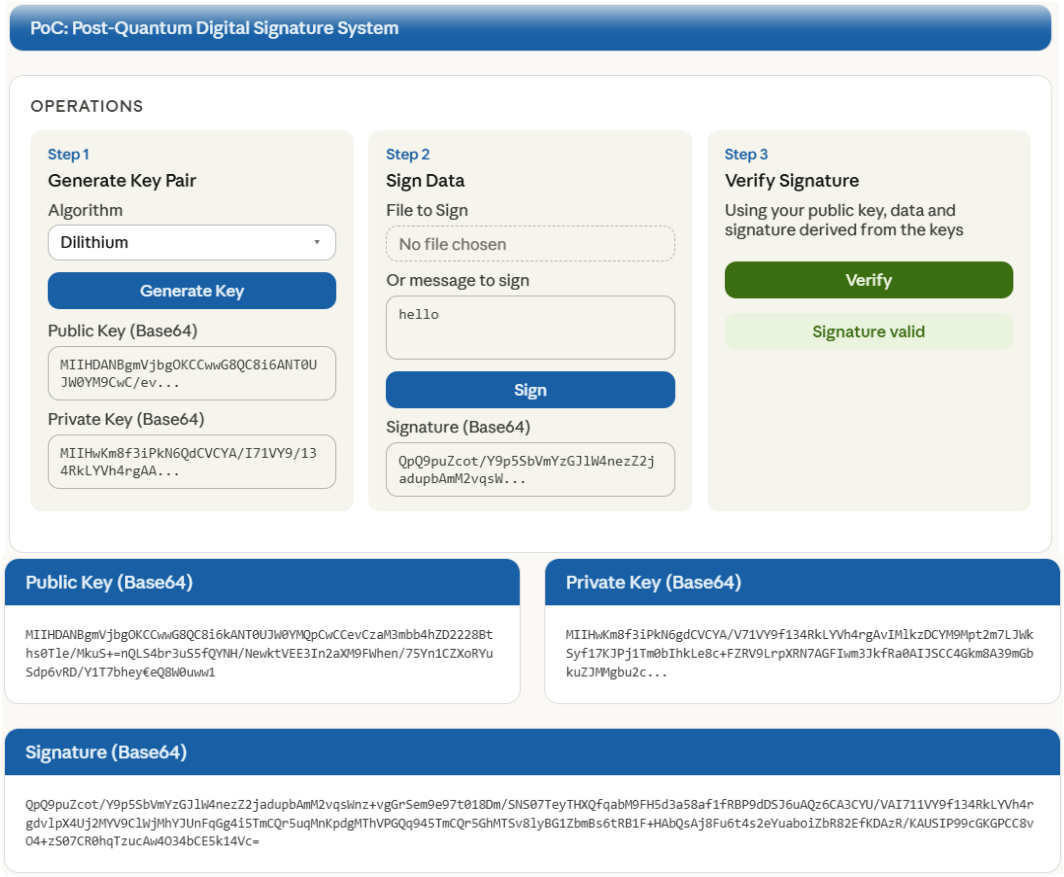


FIGURE 5. Digital Signature System Interface

The main screen of this Proof-of-Concept system displays modules for performing post-quantum digital signature operations. It is divided into four main sections: Key Pair Generation, Data Signing, Signature Verification, and Performance Benchmarking. Each module is clearly arranged to facilitate ease of use. In the Key Pair Generation section, a dropdown menu allows users to select their preferred cryptographic algorithm, such as Dilithium, SPHINCS+, RSA, or ECDSA, before clicking the "Generate Key" button. The generated key pair is then displayed in Base64 format as a reference for users or researchers as shown in Figure 6.

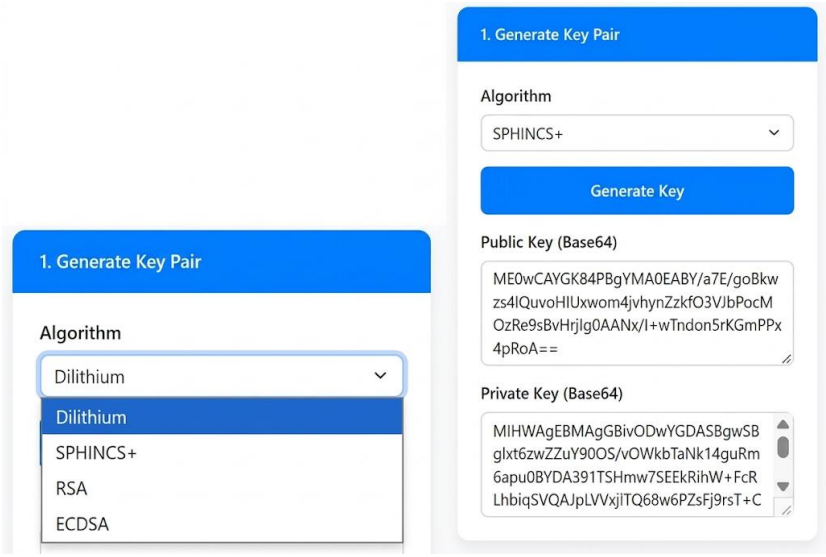


FIGURE 6. Dropdown Menu for Algorithm Selection and Key Generation

Users can sign data or messages, and the resulting digital signature is displayed in Base64 format within a text box. Using the public key, original data, and the generated signature, users can use the Signature Verification module to verify the signature's authenticity. The signature verification interface is shown in Figure 7.

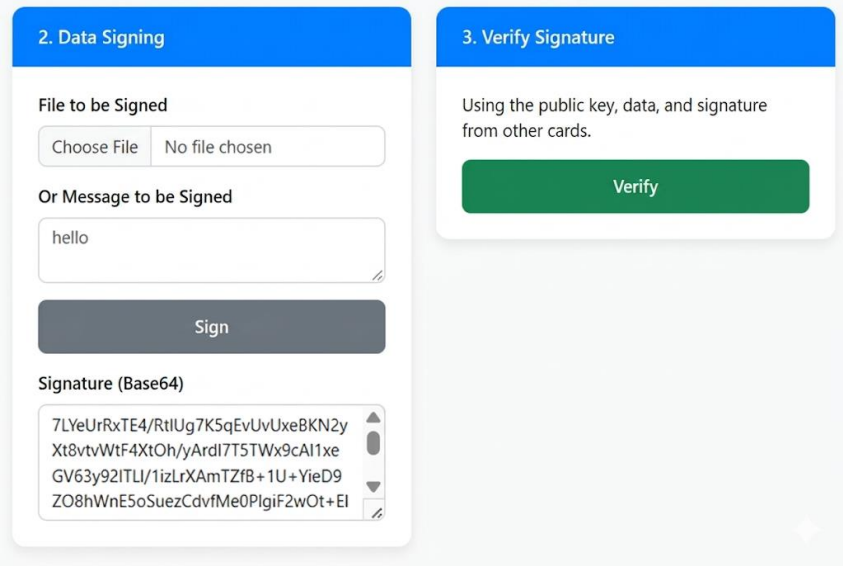
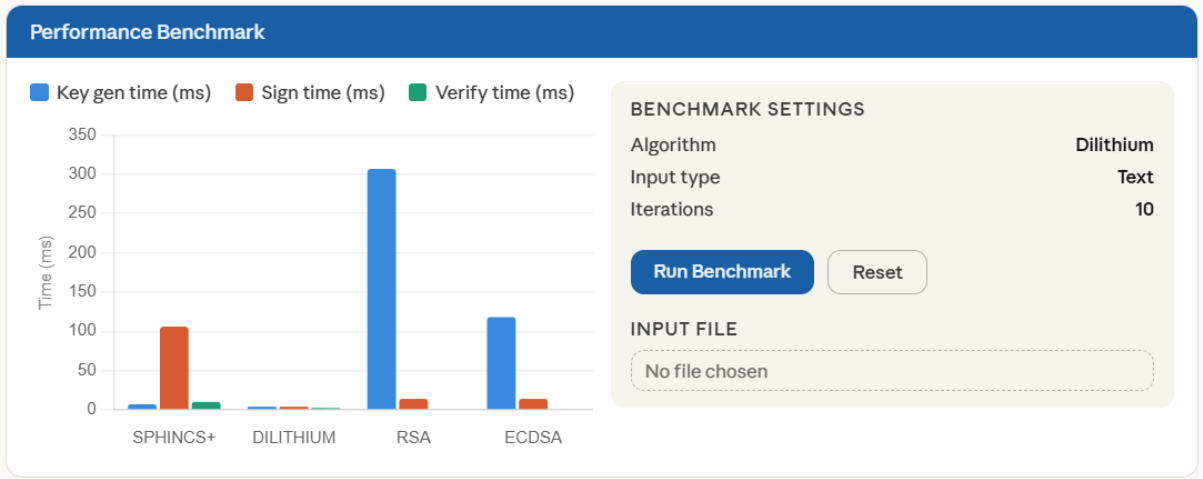


FIGURE 7. Signature Verification Interface

The Performance Benchmarking section allows users to select an algorithm, choose the input type (file or message), and upload the input to perform performance tests. After execution, results are shown in tabular format. Benchmark outcomes are also visualized using bar charts. These charts show the time required for key generation, digital signing, and verification for each algorithm, enabling users to easily compare performance. Benchmarking interface and algorithm performance chart are shown in Figure 8.



Benchmark Results

Algorithm	Type	Key gen (ms)	Sign (ms)	Verify (ms)	Public key (bytes)	Private key (bytes)	Signature (bytes)
ECDSA	Classical	118	14	0	67	—	70
RSA	Classical	307	14	0	294	1217	256
DILITHIUM	PQC	4	4	2	1976	6019	3309
SPHINCS+	PQC	7	106	10	47	118	17088

FIGURE 8. Benchmarking Interface and Algorithm Performance Chart

The backend was developed with Spring Boot in Java, while the frontend was built with Bootstrap to ensure responsiveness and simplicity. Integration of the RSA, ECDSA, CRYSTALS-Dilithium (specifically the Dilithium3 variant), and SPHINCS+ algorithms was completed using the Bouncy Castle cryptographic library. This library provided the cryptographic primitives necessary for key generation, digital signing, and signature verification. The system allowed users to dynamically select the desired algorithm and displayed benchmarking results to assist in performance evaluation. The final phase, the testing phase, played a critical role in validating both the functional and non-functional aspects of the developed digital signature system. Functional testing was performed to ensure that the system's core features, including key generation, digital signing, and signature verification, operated correctly and consistently maintained data integrity.

Each algorithm, namely RSA, ECDSA, CRYSTALS Dilithium, and SPHINCS Plus, was tested to confirm its ability to produce valid signatures and accurately verify original messages or detect any tampering. Non-functional testing was conducted to assess the system’s performance across several key metrics. These metrics included key generation time (milliseconds), signing time (milliseconds), verification time (milliseconds), key size (kilobytes), and signature size (kilobytes). Each test was repeated ten times per algorithm, and the average values were recorded for analysis. To facilitate better understanding and interpretation of test results, the system incorporated Chart.js, an open-source JavaScript library, to generate interactive, responsive bar graphs. These visualizations enabled clear comparisons across algorithms by displaying performance metrics in a user-friendly, dynamic format. This approach not only enhanced the clarity of the results but also supported developers and researchers in selecting appropriate algorithms for specific use cases. Figure 9 shows the overview of the project.

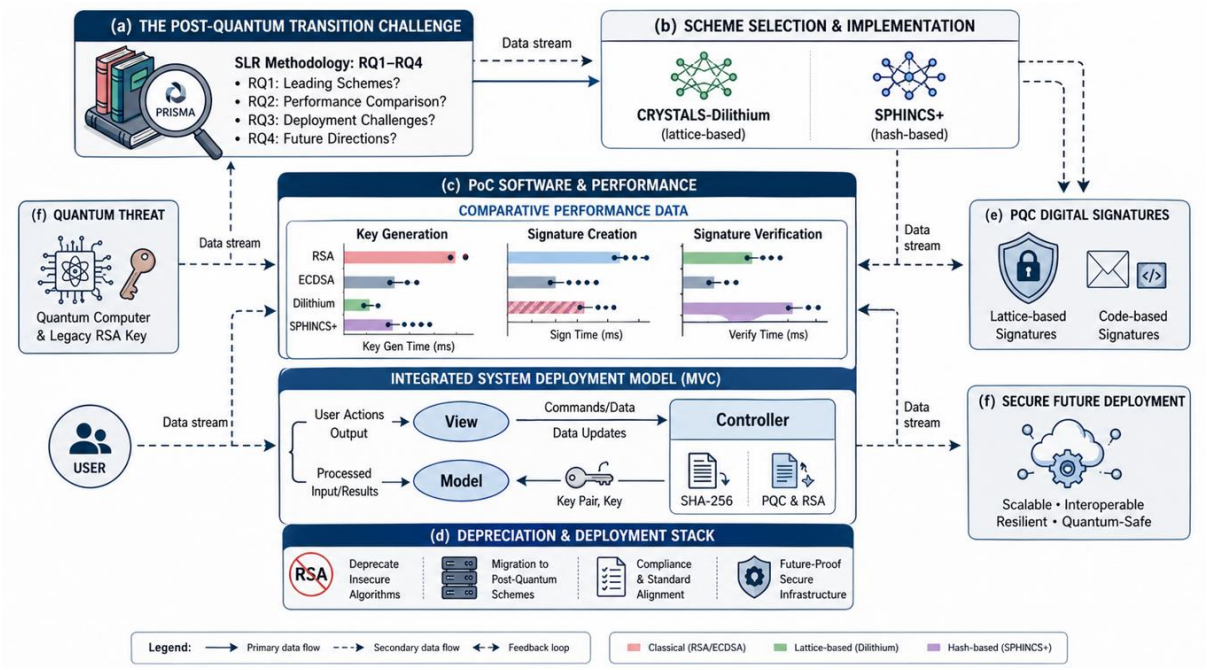


FIGURE 9. Project overview

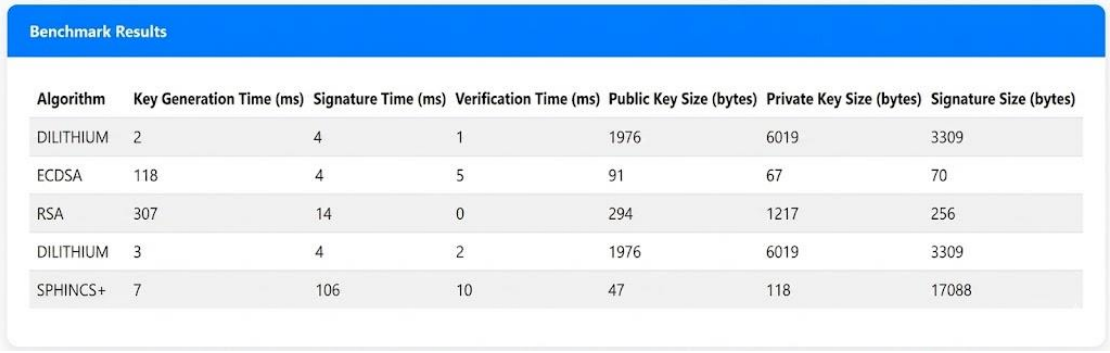
RESULTS AND DISCUSSION

This section presents a detailed analysis of the results obtained from the performance testing of the developed web-based digital signature system. The evaluation focused on four cryptographic algorithms: RSA, ECDSA, CRYSTALS-Dilithium, and SPHINCS+. Each algorithm was repeatedly tested to measure key generation time, signing time, verification time, key size, and signature size. The goal was to determine the feasibility and efficiency of post-quantum algorithms compared to conventional algorithms within a practical digital signature platform.

The experimental results reveal clear differences in performance characteristics among the tested algorithms. CRYSTALS-Dilithium emerged as the most efficient in terms of key generation and verification time. Its performance consistently outpaced RSA and ECDSA, making it suitable for applications requiring rapid response and low latency. This efficiency stems from its lattice-based design, which provides strong security without significantly compromising speed.

On the other hand, SPHINCS+ showed the slowest signing time and the largest signature size. As a stateless hash-based algorithm, SPHINCS+ prioritizes robustness and long-term security against quantum attacks. However, this comes at the cost of computational overhead and larger data footprint, making it less suitable for bandwidth-constrained or latency-sensitive environments. Despite these limitations, its ability to function without maintaining key state is a strong advantage for use cases demanding high assurance and low trust in persistent storage. RSA and ECDSA, which are widely adopted in current security infrastructures, demonstrated acceptable signing speeds and smaller signature sizes. However, RSA had the highest key-generation time, averaging over 300 milliseconds. While both algorithms remain viable in classical computing contexts, their vulnerability to quantum attacks renders them less appropriate for future-forward security systems. Their role in this project serves as a benchmark against which post-quantum algorithms were compared. To support visualization and analysis,

the system integrates Chart.js, a dynamic JavaScript library for graphically representing benchmark data, and presents a benchmark table, as shown in Figure 10.



Benchmark Results

Algorithm	Key Generation Time (ms)	Signature Time (ms)	Verification Time (ms)	Public Key Size (bytes)	Private Key Size (bytes)	Signature Size (bytes)
DILITHIUM	2	4	1	1976	6019	3309
ECDSA	118	4	5	91	67	70
RSA	307	14	0	294	1217	256
DILITHIUM	3	4	2	1976	6019	3309
SPHINCS+	7	106	10	47	118	17088

FIGURE 10. Benchmark Results

The performance results, as presented in the benchmark table, clearly demonstrate notable distinctions among the algorithms. CRYSTALS-Dilithium outperformed other algorithms in most categories, recording the fastest key generation at 2–3 milliseconds, signing at 4 milliseconds, and verification at 1–2 milliseconds. Despite its relatively large key size (1976 bytes public, 6019 bytes private) and signature size (3309 bytes), its speed and quantum resilience make it an ideal candidate for secure applications where latency is critical. ECDSA, a traditional elliptic curve-based algorithm, performed moderately well, with a key generation time of 118 ms, signing time of 4 ms, and verification time of 5 ms. It maintained a small key footprint (91 bytes public, 67 bytes private) and the smallest signature size at only 70 bytes, making it suitable for environments with strict memory or transmission constraints. However, its vulnerability to quantum computing attacks limits its long-term viability.

RSA, although widely deployed in current cryptographic systems, had the highest key-generation time of 307 milliseconds, indicating a significant computational load during setup. However, it achieved instant verification (0 ms) and reasonable signing time (14 ms). With moderate key sizes (294 bytes public, 1217 bytes private) and a 256-byte signature, RSA remains practical for legacy systems but is less suited for future quantum-safe applications. SPHINCS+, a stateless hash-based post-quantum signature scheme, showed a distinct trade-off between security and performance. While its key sizes were exceptionally small (47 bytes public, 118 bytes private), it had the slowest signing time at 106 milliseconds and the largest signature size at 17088 bytes. This large signature size may pose challenges in bandwidth-limited or storage-constrained systems, although the algorithm provides strong long-term security.

Among all the algorithms tested, CRYSTALS Dilithium demonstrated the most efficient performance in terms of key generation and verification time, as shown in Figure 11. It significantly outperformed RSA and ECDSA in these aspects, making it a strong candidate for applications that require speed and low-latency verification. On the other hand, SPHINCS Plus, although providing a high level of post-quantum security, showed the slowest signing time and produced the largest signature sizes. This indicates a trade-off between robustness and operational efficiency, especially in storage-limited or real-time environments. Overall, the testing phase confirmed that the system is functionally stable and meets the necessary performance requirements, making it suitable for future expansion or deployment within secure digital environments.

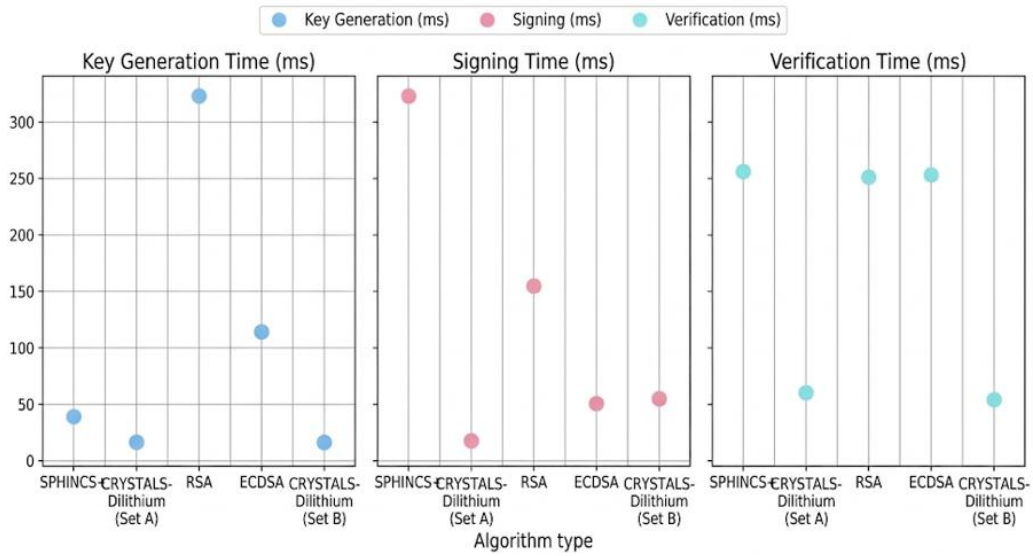


FIGURE 11. Comprehensive multi-metric cryptographic signature evaluation

In summary, the performance tests demonstrate that post-quantum algorithms such as CRYSTALS-Dilithium can outperform classical algorithms like RSA and ECDSA in both speed and quantum resilience, albeit with slightly larger keys and signatures. SPHINCS+, while offering excellent security properties, incurs significant computational and storage overhead. These insights confirm the system's capability to evaluate digital signature schemes under practical conditions and support informed decision-making in selecting suitable algorithms for secure digital infrastructures.

FUTURE WORK

Although the developed digital signature system successfully integrates classical and post-quantum cryptographic algorithms, there is still significant potential for further enhancement. One of the key directions for future work is the integration of X.509 digital certificates and PKCS#12 key storage format, which would allow the system to function within a standard Public Key Infrastructure (PKI). This would increase its compatibility with real-world enterprise environments where identity verification and trust chains are critical. In addition, the system could benefit from a more robust key and certificate management mechanism. This includes secure encryption and storage of private keys, public keys, and digital certificates in a scalable database, along with implementation of access controls, logging, and auditing features to meet security compliance standards. Incorporating certificate revocation features such as Certificate Revocation Lists (CRL) or the Online Certificate Status Protocol (OCSP) would also enhance the system's trust model by allowing users to verify the validity of certificates in real time.

Further improvements may include support for additional post-quantum cryptographic algorithms, such as Falcon or Classic McEliece, which have also been recommended by NIST. Tuning the parameters of existing algorithms could help optimize the balance between performance and security, especially for use in resource-constrained environments. Moreover, the user interface can be refined to offer better user experience through real-time feedback, accessibility enhancements, and multi-language support, making the platform more inclusive and easier to use. Expanding the system to support mobile and cross-platform deployment would also increase its practicality, enabling users to sign and verify documents securely on

smartphones and tablets. This would be particularly useful in domains such as e-government, healthcare, and financial services where portability is important. Finally, future development should also include formal verification of the cryptographic components and external security audits to ensure the system's resilience and trustworthiness before it is deployed in critical environments. These enhancements will help elevate the proof-of-concept system into a fully functional and secure digital signature solution, ready for adoption in real-world applications that require long-term resistance to quantum threats.

CONCLUSION

The paper successfully developed a proof-of-concept web-based digital signature system that integrates both traditional and post-quantum cryptographic algorithms. This paper contributes a comprehensive crypto-agile framework that transitions digital signature systems from legacy RSA and ECDSA standards to NIST-standardized post-quantum algorithms, specifically CRYSTALS-Dilithium and SPHINCS+. By integrating a functional Model-View-Controller (MVC) based Proof-of-Concept with a dedicated PQC/RSA Bridge Layer, the paper demonstrates a practical implementation path that maintains backward compatibility while securing infrastructure against future quantum threats. Furthermore, the research provides an empirical performance baseline through rigorous benchmarking of key generation and verification latencies, offering a strategic deployment roadmap that addresses hardware, protocol, and policy requirements for critical data environments. The benchmarking results indicate that while RSA and ECDSA offer compact signatures and fast performance, they are vulnerable to quantum attacks. CRYSTALS-Dilithium presents a balanced solution with reasonable security and performance, making it a practical candidate for near-future adoption. SPHINCS+, although slower and with a larger signature size, provides the highest level of post-quantum assurance. This paper affirms that integrating PQC algorithms into digital signature systems is not only technically feasible but also essential to future-proofing digital security infrastructure against quantum threats.

ACKNOWLEDGEMENT

This work was supported by the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, and MSC Trusgate under Grant Scheme TT-2024-018.

REFERENCES

Astarloa, Armando, Jesus Lazaro, and Jose Ignacio Garate. "CRYSTALS-Dilithium Post-Quantum Cyber-Secure SoC for Wired Communications in Critical Systems." Internet of Things 33 (September 2025): Art. 101656. https://doi.org/10.1016/j.iot.2025.101656.

Bos, Joppe W., Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS – Kyber and Dilithium in Practice." In Advances in Cryptology – EUROCRYPT 2018, edited by Jesper Buus Nielsen and Vincent Rijmen, 327–346. Cham: Springer, 2018.

Carril, Xavier, Charalampos Kardaris, Jordi Ribes-Gonzalez, Oriol Farras, Carles Hernandez, Vatistas Kostalabros, Joel Ulises Gonzalez-Jimenez, and Miquel Moreto. "Hardware Acceleration for High-Volume Operations of CRYSTALS-Kyber and CRYSTALS-Dilithium." ACM Transactions on Reconfigurable Technology and Systems 17, no. 3 (September 2024): Art. 41. https://doi.org/10.1145/3675172.

Chen, Lily, Lily-Kuo Chen, Scott Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on Post-Quantum Cryptography. U.S. Department of Commerce, NIST, 2016.

Dorota, Daria. "An Overview of Cryptography and Network Security." International Journal of Advanced Trends in Computer Science and Engineering 12, no. 1 (2023): 1–9.

Hülsing, Andreas, Joost Rijneveld, Ralf Künnemann, Panos Kampanakis, and Thomas Wiggers. "SPHINCS+ Specification v3.1." 2022. https://sphincs.org/data/sphincs+-r3.1-specification.pdf.

Moody, Dustin, Ray Perlner, Andrew Regenscheid, Andrew Robinson, and David Cooper. Transition to Post-Quantum Cryptography Standards. NIST Internal Report (NISTIR) 8547 (Draft). National Institute of Standards and Technology, 2024.

NIST. "NIST Announces First Post-Quantum Cryptography Standards." August 13, 2024. https://www.nist.gov/news-events/news/2024/08/nist-announces-pqc-standards.

PQ-Crystals. "Kyber Specification (Round 3)." 2021. https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf.

Post-Quantum. "NIST's PQC Technical Standardization: What You Need to Know." 2025. https://postquantum.com/post-quantum/nists-pqc-technical.

Red Hat. "Post-Quantum Cryptography: Understanding the Future of Security." Red Hat Blog, 2023. https://www.redhat.com/en/blog/post-quantum-cryptography.

Shin, Donghyun, Youngbeom Kim, and Seog Chung Seo. "Optimizing Crystals-Dilithium Implementation in 16-bit MSP430 Environment Utilizing Hardware Multiplier." ICT Express 11, no. 1 (February 2025): 59–65. https://doi.org/10.1016/j.icte.2024.09.019.

Wang, M., and G. L. Long. "Lattice-Based Access Authentication Scheme for Quantum Communication Networks." Science China Information Sciences 67, no. 12 (2024): 222501.

Wang, Ziheng, Xiaoshe Dong, Heng Chen, Yan Kang, and Qiang Wang. "CUSPX: Efficient GPU Implementations of Post-Quantum Signature SPHINCS+." IEEE Transactions on Computers 74, no. 1 (January 2025): 15–28. https://doi.org/10.1109/TC.2024.3457736.

Wu, Jiafei, Yifei Yu, Zhao Chen, Hao Yang, Chao Li, and Zhe Liu. "CBPSPX: A CUDA-Based Batch Parallel Optimization of Post-Quantum Signature SPHINCS+." IEEE Internet of Things Journal 12, no. 18 (September 2025): 37898–911. https://doi.org/10.1109/JIOT.2025.3585558.

Zachary, M. Z., S. Sylviani, and E. Kurniadi. "Implementasi Algoritma RSA (Rivest-Shamir-Adleman) pada Kriptografi Klasik." Mathematical Sciences and Applications Journal 4, no. 2 (2024): 54–59. https://doi.org/10.22437/msa.v4i2.28863.