Post-Quantum Secure Email Signatures: DKIM Signing with CRYSTALS-Dilithium

Swapnil Garg 22115150
Mmukul Khedekar 22114054

Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee - 247667 (INDIA)

Supervisor : Dr. Manoj Misra
Supervisor : Dr. Sanjeev Shukla

May, 2025

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE, ROORKEE

Candidates' Declaration

We hereby certify that the work which is being presented in this report entitled "Post-Quantum Secure Email Signatures: DKIM Signing with CRYSTALS-Dilithium" in partial fulfillment of the requirements for the award of the degree of Bachelor Of Technology and submitted in the Department of Computer Science and Engineering of the Indian Institute of Technology Roorkee is an authentic record of our own work carried out during the period from January, 2025 to May, 2025 under the supervision of Dr. Manoj Misra, Professor and Dr. Sanjeev Shukla, Senior Scientific Officer.

Swapnil Garg **22115150**
Mmukul Khedekar **22114054**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: May, 2025

Supervisor
(Dr. Manoj Misra)

Supervisor
(Dr. Sanjeev Shukla)

Abstract

The transition to post-quantum cryptography is an emerging challenge as quantum computing threatens the security of classical public-key cryptographic schemes. In this project, we explore the integration of CRYSTALS-Dilithium, a NIST-selected post-quantum digital signature algorithm, into the DomainKeys Identified Mail (DKIM) framework used for checking email integrity. While DKIM traditionally relies on RSA or elliptic curve signatures, these are vulnerable to quantum attacks. Our objective is to demonstrate the feasibility of using Dilithium to sign and verify email messages within the DKIM model, without modifying the fundamental structure of DKIM. To support this, we developed a complete experimental infrastructure including a user agent, SMTP relay, and a DNS server capable of publishing Dilithium public keys via TXT records. The implementation highlights the practical considerations of using post-quantum signatures in real-world protocols, such as handling larger key and signature sizes. The focus of this project is limited to tackling the cryptographic unforgeability of signatures; issues such as email phishing and social engineering are out of the scope of this project.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

Email is one of the most widely used communication protocols in the modern world. However, it was originally designed without built-in security mechanisms for verifying the sender's identity or ensuring message integrity. This has made email susceptible to several types of abuse, including spoofing, phishing, and spam.

To address these issues, standards such as **DomainKeys Identified Mail (DKIM)** have been introduced. DKIM allows a domain to sign its outgoing emails using a digital signature, which can be verified by recipient servers using the domain's public key published in DNS. This ensures that the message has not been altered in transit and that it was authorized by the domain.

DKIM currently relies on classical digital signature schemes, primarily **RSA** and **Elliptic Curve Cryptography (ECC)**. While these schemes are secure against classical adversaries, they are fundamentally threatened by advances in quantum computing. In particular, the development of **Shor's algorithm** has demonstrated that quantum computers can efficiently solve the underlying hard problems—*integer factorization* for RSA and *discrete logarithm* for ECC—rendering these schemes insecure in a post-quantum world.

1.2 Quantum Threats to Classical Cryptography

Quantum computers, once they reach sufficient scale and stability, will have the capability to break widely deployed public-key cryptographic algorithms. The two most relevant quantum algorithms are:

- **Shor's Algorithm** [5]: Provides an efficient method to factor large integers and compute discrete logarithms in polynomial time. This directly compromises:
  - **RSA**, which relies on the difficulty of factoring large integers.
  - **DSA** and **DH**, based on the discrete logarithm problem.

  - **ECC**, based on the elliptic curve discrete logarithm problem.

  If a quantum computer runs Shor's algorithm, it can derive the private key from any exposed public key, breaking digital signature schemes and enabling forgeries.

- **Grover's Algorithm** [1]: Offers a quadratic speedup for brute-force search problems, impacting symmetric key algorithms by effectively halving their security level. While Grover's algorithm does not directly compromise DKIM's asymmetric signatures, it emphasizes the need for stronger cryptographic primitives across protocols.

For DKIM, which depends on the assumption that the private key cannot be derived from the public key, **Shor's algorithm presents a critical vulnerability**, motivating the need for post-quantum secure signature schemes.

1.3 Post-Quantum Cryptography and Dilithium

Post-Quantum Cryptography (PQC) aims to develop cryptographic algorithms that remain secure even in the presence of large-scale quantum computers. These algorithms rely on mathematical problems that are believed to be hard for both classical and quantum adversaries.

Among the promising candidates is **lattice-based cryptography**, particularly the **CRYSTALS-Dilithium** digital signature scheme. Dilithium is based on hard problems in lattices, such as the *Short Integer Solution (SIS)* and *Learning With Errors (LWE)* problems, which currently have no known efficient quantum algorithms to solve them. This makes Dilithium a leading candidate for post-quantum secure digital signatures and has led to its standardization by NIST.

1.4 Problem Statement

While DKIM ensures email authentication, its reliance on RSA and ECC makes it vulnerable to quantum attacks. With practical quantum computers, adversaries could forge DKIM signatures using Shor's algorithm.

This project explores integrating **CRYSTALS-Dilithium** into the DKIM signing and verification process to replace RSA and achieve post-quantum security. Our aim is to ensure email signatures remain unforgeable against quantum adversaries, safeguarding email integrity and authenticity.

We developed a complete email system supporting both RSA and Dilithium signatures to compare their integration into the DKIM workflow. The focus is solely on cryptographic unforgeability; other threats like phishing or social engineering are out of scope.

Chapter 2

Methodology

To evaluate the integration of CRYSTALS-Dilithium into the DKIM[3] email authentication framework, we developed a custom, end-to-end email communication system. This setup was designed to give us complete control over the message flow, cryptographic operations, and key distribution mechanisms, enabling precise experimentation with post-quantum signature schemes in a real-world protocol context. Additionally, we have also implemented CRYSTALS-Dilithium referring closely to the specifications in the final standard published by **NIST as FIPS 204**[4].

2.1 System Architecture

The system comprises four key components:

- **Mail User Agent (MUA)**: A simple interface for composing and sending email messages, responsible for preparing the message before it is handed off for transmission.
- **SMTP Client and Server**[2]: The SMTP Client and Server handle the emailing connection along with generating signatures for the email and verifying them at the client and server, respectively.
- **DNS server**: A custom DNS server that handles queries for Domain to IP DNS lookup requests during SMTP connections and TXT records containing DKIM public keys, supporting key formats for various different signing schemes.
- **Signature and verification algorithms**: The actual algorithms used for signing and verification of emails, which include both existing approaches and Dilithium.

By implementing each of these components from scratch, we avoided dependencies on existing infrastructure and ensured that the cryptographic substitution could be studied in isolation. The dual support for traditional and post-quantum algorithms allows us to directly compare their behavior and performance in a uniform test environment.

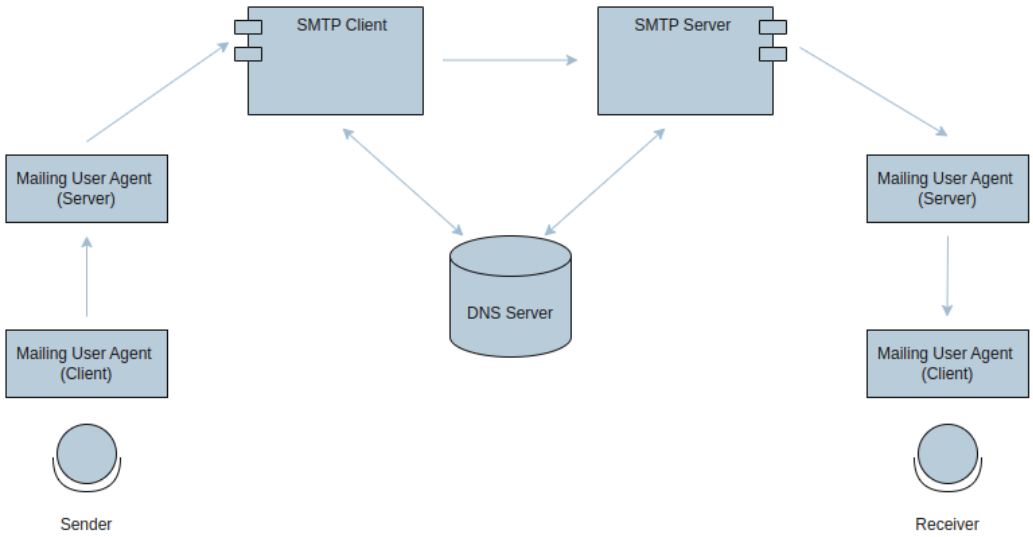

FIGURE 2.1: High-Level System Architecture

2.2 Implementing CRYSTALS-Dilithium

This implementation aligns with the parameters and design principles established in the **final specification published by NIST as FIPS 204** for the post-quantum cryptographic standardization. Our implementation ensures compatibility with the standardized version of Dilithium under FIPS 204 and guarantees that our implementation meets the security and performance benchmarks set forth by NIST. The CRYSTALS-Dilithium algorithm is structured into three main procedures: key generation, signing, and verification.

2.2.1 Key Generation

Algorithm 1 Gen

1: $\mathbf{A} \leftarrow R_q^{k \times \ell}$
2: $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow S_\eta^\ell \times S_\eta^k$
3: $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
4: **return** $(\mathsf{pk} = (\mathbf{A}, \mathbf{t}), \mathsf{sk} = (\mathbf{A}, \mathbf{t}, \mathbf{s}_1, \mathbf{s}_2))$

2.2.2 Signing Procedure

Algorithm 2 Sign(sk, M)

$\mathbf{z} := \perp$
while $\mathbf{z} = \perp$ **do**
  $\mathbf{y} \leftarrow S^{\ell}_{\gamma_1 - 1}$
  $\mathbf{w}_1 := \text{HighBits}(\mathbf{Ay}, 2\gamma_2)$
  $c \in \mathcal{B}_{60} := H(M \,\|\, \mathbf{w}_1)$
  $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$
  if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ **or** $\|\text{LowBits}(\mathbf{Ay} - c\mathbf{s}_2, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$ **then**
    $\mathbf{z} := \perp$
  end if
end while
return $\sigma = (\mathbf{z}, c)$

2.2.3 Verification Procedure

Algorithm 3 Verify(pk, $M, \sigma = (\mathbf{z}, c)$)

$\mathbf{w}'_1 := \text{HighBits}(\mathbf{Az} - c\mathbf{t}, 2\gamma_2)$
if $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$ **and** $c = H(M \,\|\, \mathbf{w}'_1)$ **then**
  return Valid
else
  return Invalid
end if

2.3 Implementing Existing Standards

For the purpose of accurate and meaningful comparisons between CRYSTALS-Dilithium and current industry-standard digital signature algorithms, we selected three of the most widely adopted schemes: RSA, ECDSA and Ed25519.

We implemented each of these algorithms within our custom DKIM-compatible email system to provide a consistent framework for comparison. This implementation allowed us to evaluate key metrics such as signature size, verification speed, and integration complexity when transitioning to a post-quantum secure algorithm like Dilithium. By running all schemes under the same infrastructure, we ensured a fair assessment of their respective suitability for future-proof email authentication.

Chapter 3

Results

After implementing the emailing infrastructure and the CRYSTALS-Dilithium digital signature scheme, we conducted comprehensive benchmarking of the complete system. Each component of the signature algorithm was tested across varying parameters to analyze performance trends.

To provide a meaningful comparison, we evaluated three variants of the Dilithium scheme, namely the Dilithium 2, 3, and 5, against three widely deployed classical digital signature algorithms: RSA-SHA256 (the most commonly used), ECDSA, and ED25519. These classical algorithms are expected to be vulnerable in a post-quantum world, making such a comparative study vital for post-quantum cryptography adoption.

1. Key Generation Time of Different Algorithms

We evaluated the efficiency of key generation for each algorithm by executing 10,000 iterations. The mean execution time and standard deviation were computed, and the results are presented as bar graphs and box plots for comparison.

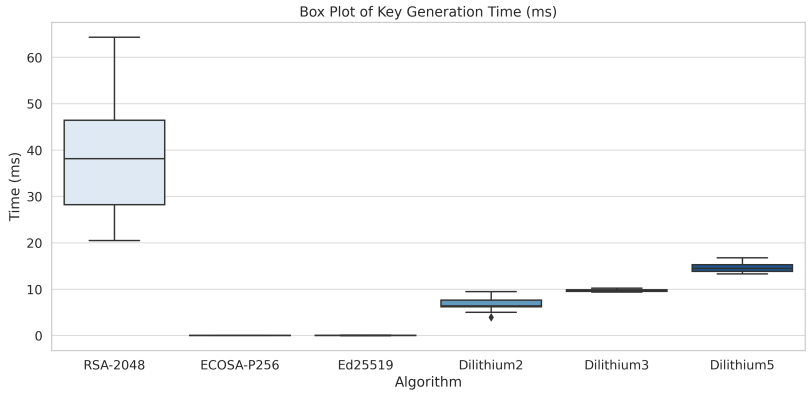

FIGURE 3.1: Distribution of key-generation times (box plots) for each digital signature algorithm.

2. Signature Generation Time of Different Algorithms

A dataset of 10,000 plaintext messages of the same size was generated and signed using each algorithm under evaluation. The mean and standard deviation of the signature generation time were calculated to assess performance variability.

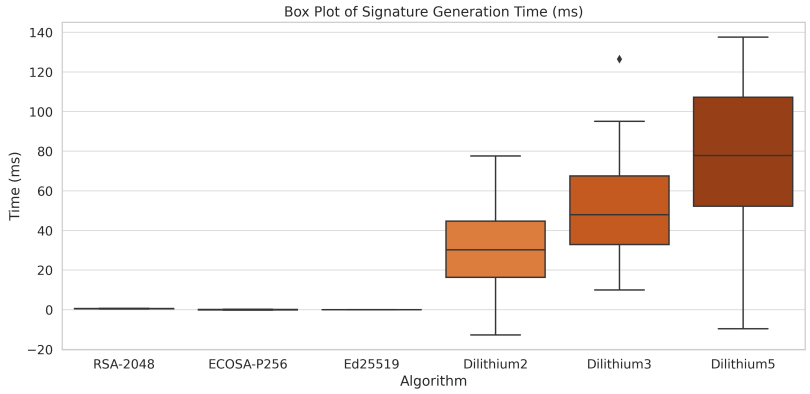

FIGURE 3.2: Statistical distribution (box plots) of signature-generation times across algorithms.

We then generated a dataset of varying message sizes and then signed them using each algorithm and computed the mean and standard deviation of the signature generation time.

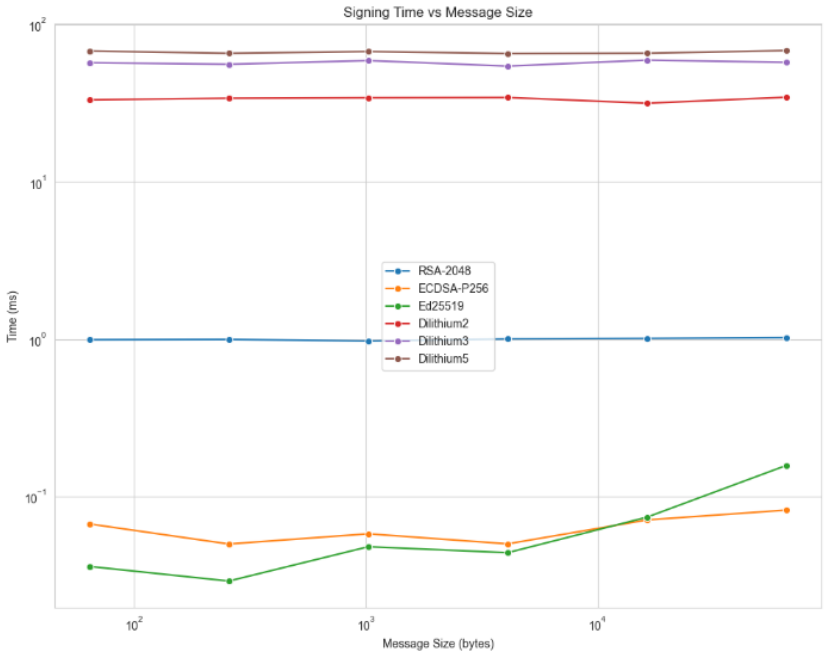

FIGURE 3.3: Mean signature-generation time for versus varying message sizes.

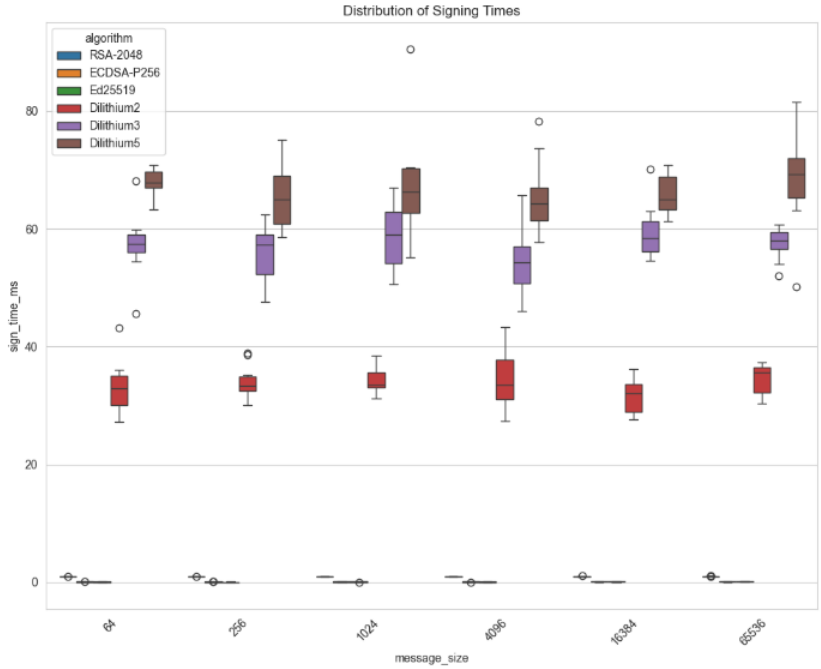

FIGURE 3.4: Statistical distribution (box plots) of signature-generation times for various message sizes.

3. Signature Size of Different Algorithms

Using the dataset from the preceding experiment, the signature sizes produced by each algorithm were measured and compared. The results are presented in the bar graph below.

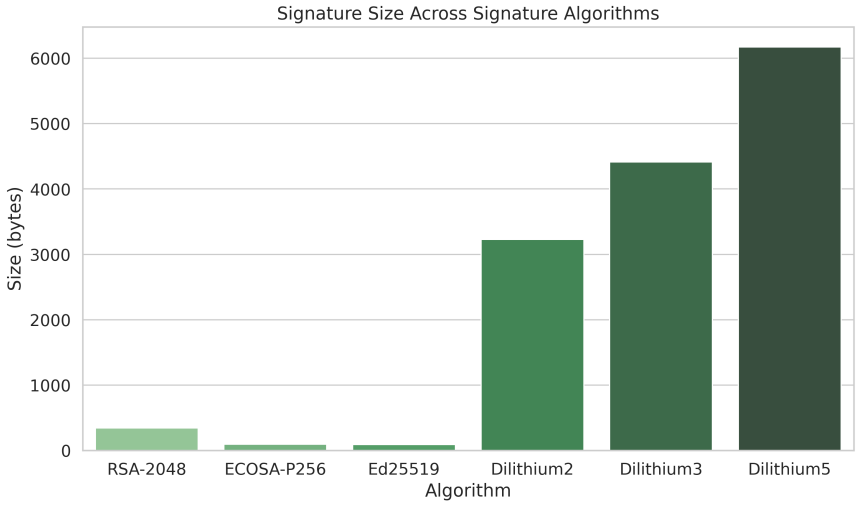

FIGURE 3.5: Signature size comparison across evaluated digital signature algorithms.

We then measured the signature sizes of the signature generated for dataset of varying sizes of the plaintext.

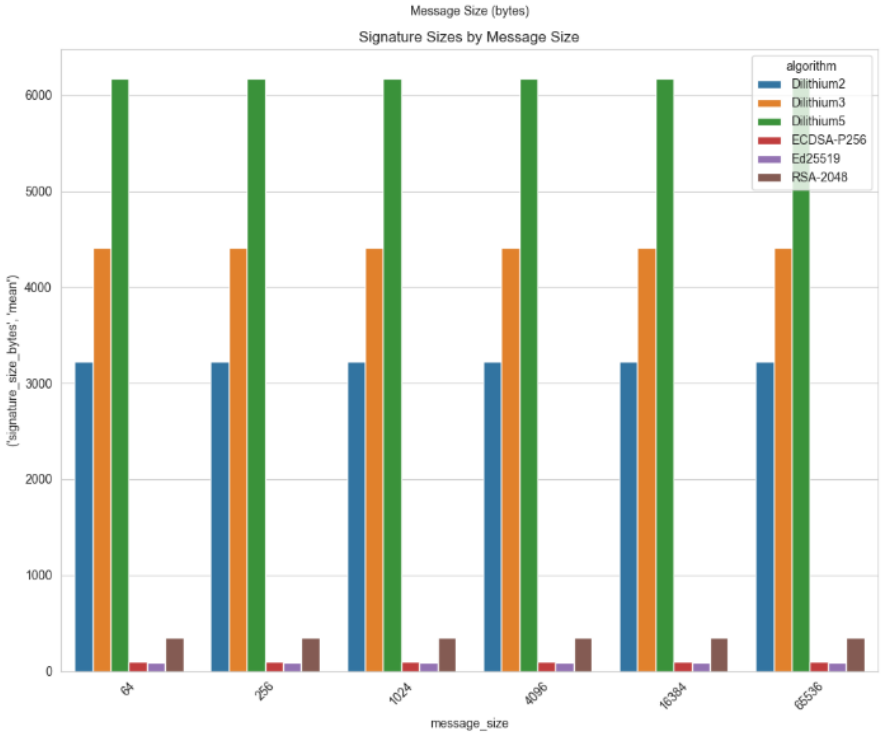

FIGURE 3.6: Signature size comparison across evaluated digital signature algorithms.

4. Verification Time of Different Algorithms

Using the signatures produced in our earlier experiments, we verified them with their corresponding algorithm, recorded the elapsed times, and calculated the mean and standard deviation to generate the accompanying box plot.

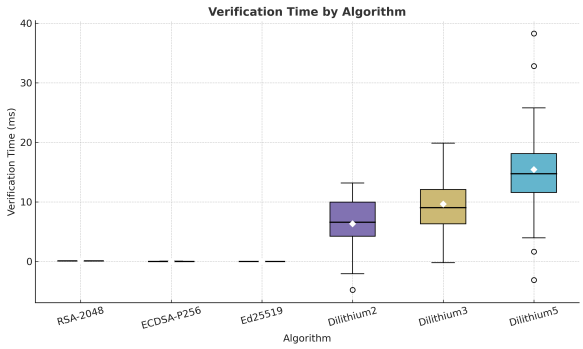

FIGURE 3.7: Signature size comparison across evaluated digital signature algorithms.

Conclusion

Our evaluation demonstrates that the Dilithium key-generation procedure outperforms RSA 2048 primarily because RSA requires additional constraints to ensure safe prime selection—yet remains slower than modern elliptic-curve schemes such as ECDSA-P256 and Ed25519. However, the public and private keys produced by Dilithium are substantially larger than those of the classical algorithms, which in practice could translate into significantly larger DNS TXT records and increased storage requirements.

In terms of signing performance, Dilithium exhibits markedly higher computation times than all classical signature schemes tested. Moreover, signature-generation latency increases with the security level (Dilithium-2 $\leq$ Dilithium-3 $\leq$ Dilithium-5), and this hierarchy persists regardless of message size. Although the absolute signing times did not vary appreciably with different message lengths, the relative ordering of algorithms remained consistent with the fixed-message benchmarks.

Signature sizes for Dilithium are also considerably larger, by an order of magnitude in some cases, leading to bulkier DKIM headers in e-mail applications. This characteristic represents a practical drawback as e-mails signed with Dilithium will incur greater bandwidth and storage costs. As with signing time, signature length proved stable across all tested message sizes.

Finally, verification costs follow the same trend: Dilithium verification routines are significantly more computationally intensive than their classical counterparts, with higher-security variants imposing the greatest overhead. Thus, while Dilithium offers the critical advantage of post-quantum security, its performance characteristics, especially larger key and signature sizes and slower signing and verification must be carefully weighed in real-world deployments.

Bibliography

[1] Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996. URL https://arxiv.org/abs/quant-ph/9605043.

[2] Dr. John C. Klensin. Simple Mail Transfer Protocol. RFC 5321, October 2008. URL https://www.rfc-editor.org/info/rfc5321.

[3] Murray Kucherawy, Dave Crocker, and Tony Hansen. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, September 2011. URL https://www.rfc-editor.org/info/rfc6376.

[4] National Institute of Standards and Technology. Module-lattice-based digital signature standard. Federal Information Processing Standards Publication FIPS 204, U.S. Department of Commerce, Washington, D.C., 2024. URL https://doi.org/10.6028/NIST.FIPS.204.

[5] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi: 10.1109/SFCS.1994.365700.