

Article

From Algebraic Correctness to Zero Trust Deployment: An Assurance Framework for ML-KEM

William Edwards¹, Miroslav Vukovic² and Jeffrey Wallace^{2,*}¹ Deep Blue Krypto Group, Overland Park, KS 66213, USA; will.edwards@deepbluekrypto.com² Black Swan Syndicate LLC, Washington, DC 20006, USA; miro@bss.dev

* Correspondence: jeffrey@bss.dev; Tel.: +1-689-247-0622

Abstract

The transition from post-quantum cryptographic standardization to operational deployment requires more than the selection of a quantum-resistant algorithm. It requires traceability from the mathematical assumptions of the primitive to implementation requirements, protocol composition, migration controls, and runtime governance. This paper develops a cross-layer assurance framework for deploying the NIST-standardized Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) within crypto-agile Zero Trust architectures. The paper presents a simplified algebraic model of the public-key encryption operations underlying ML-KEM, emphasizing quotient-ring arithmetic, module operations, controlled noise, and cancellation of the principal bilinear term. It then distinguishes this explanatory model from the complete mechanism specified in FIPS 203, including standardized sampling, encoding, compression, hashing, key derivation, ciphertext consistency checking through re-encryption and comparison, implicit rejection, and prescribed decapsulation behavior. The principal contribution is an assurance framework connecting three levels: algebraic assurance, implementation assurance, and cryptographic governance. A deployment architecture, threat model, crypto-agility lifecycle, and bounded AI-assisted monitoring model are presented to show how ML-KEM profiles can be inventoried, approved, negotiated, observed, migrated, rolled back, and audited without altering the underlying cryptographic guarantees. The resulting framework provides a technically grounded bridge between ML-KEM mathematics and practical post-quantum migration in Zero Trust systems.

Keywords: ML-KEM; post-quantum cryptography; module lattices; zero trust architecture; crypto agility; cryptographic assurance; cryptographic governance; hybrid key establishment



Academic Editors: Haibo Cheng and Wenting Li

Received: 24 June 2026

Revised: 20 July 2026

Accepted: 22 July 2026

Published: 3 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

A key-encapsulation mechanism enables two parties to establish shared keying material over a public channel without transmitting that material directly. ML-KEM is a post-quantum key-encapsulation mechanism standardized in FIPS 203 that may serve as a key-establishment component within a Zero Trust architecture [1–3]. Its security is based on the hardness of the Module Learning With Errors (Module-LWE) problem, while its efficient realization combines linear algebra over modules with polynomial arithmetic in a structured quotient ring. Zero Trust does not alter the ML-KEM primitive or create a separate class of “Zero Trust cryptography.” Rather, it supplies the identity, policy, enforcement,

monitoring, and governance environment in which an approved ML-KEM implementation can be deployed.

At the algebraic level, the construction begins with a noisy public linear relation. The legitimate receiver possesses a short secret vector, whereas an adversary observes a matrix relation perturbed by sampled error terms. The underlying encryption operation introduces ephemeral randomness and additional noise so that the receiver can cancel the principal structured contribution during decryption, leaving an embedded message together with a residual error. This cancellation provides useful correctness intuition, but it represents only one layer of ML-KEM. The equations developed in this paper are therefore identified throughout as a simplified algebraic model of the public-key encryption operations underlying ML-KEM. They are not a complete specification and must not be used as an implementation substitute for FIPS 203. A conforming implementation must execute the complete standardized mechanism, including deterministic derivation, sampling, encoding, compression and decompression, hashing, key derivation, ciphertext processing, validation, implicit rejection, and defined decapsulation behavior [1].

Existing treatments commonly address the relevant layers separately. Cryptographic standards and technical specifications define ML-KEM and its security requirements; mathematical expositions explain lattice and module arithmetic; implementation guidance addresses side channels, randomness, validation, and protocol composition [4–7]; and post-quantum migration guidance focuses on discovery, inventory, policy, and phased replacement [8–10]. This separation can obscure how a mathematical dependency becomes an implementation obligation and, subsequently, a governance control in a live system.

This paper does not propose a new KEM, parameter set, security reduction, or modification of FIPS 203. Its original contribution is an integrative assurance framework that connects three levels:

1. Algebraic assurance, covering the quotient-ring and module operations, controlled-noise model, and cancellation identity that provide correctness intuition;
2. Implementation assurance, covering standards conformity, sampling, compression, validation, implicit rejection, constant-time execution, side-channel resistance, randomness quality, and secure protocol composition [4–7];
3. Governance assurance, covering cryptographic discovery, profile approval, negotiation, inventory, monitoring, migration, rollback, retirement, and audit within a Zero Trust architecture.

The paper makes four specific contributions. First, it presents an accessible but formally stated derivation of the algebraic cancellation underlying the simplified encryption model. Second, it establishes a clear boundary between that pedagogical model and the complete FIPS 203 mechanism. Third, it introduces a Zero Trust deployment architecture and crypto-agility lifecycle that trace cryptographic dependencies into enforceable operational controls. Fourth, it defines a bounded role for AI-assisted monitoring that improves cryptographic observability without permitting an AI component to alter algorithms, keys, parameters, validation behavior, or security-critical computations.

The remainder of the paper is organized as follows. Section 2 reviews related work, identifies the research gap, and describes the analytical method. Sections 3–8 develop the algebraic setting, simplified encryption model, quotient-ring arithmetic, and module-lattice correctness perspective. Section 9 distinguishes the simplified algebraic model from the complete ML-KEM standardized in FIPS 203 and examines implementation and protocol-composition risks. Section 10 presents the proposed Zero Trust deployment architecture, threat model, crypto-agility lifecycle, protocol workflow, control mappings, and migration scenario. Section 11 defines the role, authority, data boundaries, and safeguards of AI-assisted cryptographic governance. Section 12 presents the results and discusses the

three-layer assurance framework. Section 13 identifies the limitations of the study, and Section 14 concludes the paper.

2. Related Work and Research Gap

The foundations of ML-KEM derive from the CRYSTALS-Kyber construction and its supporting security analysis, while FIPS 203 defines the interoperable standardized key-encapsulation mechanism [1,2,11]. These sources specify the complete algorithm, parameter sets, encoding and compression operations, key derivation, ciphertext processing, and decapsulation requirements. NIST SP 800-227 complements the algorithm specification by providing broader recommendations for the secure implementation and use of KEMs [4]. They are authoritative implementation references rather than substitutes for higher-level deployment architecture.

A second body of work explains Learning With Errors, ring- and module-lattice structures, quotient-ring arithmetic, and the correctness behavior of Kyber-derived encryption. Such treatments establish the mathematical context for the noisy public relation and the cancellation of structured terms during decryption. Analyses of decryption-failure probability further examine the effects of sampled error distributions, compression, and parameter-specific behavior [11,12]. The simplified derivation in this paper draws on that established foundation and does not claim a new correctness proof.

A third body of work addresses post-quantum migration. NIST, CISA, and NSA guidance emphasizes cryptographic discovery, inventory, vendor coordination, phased replacement, interoperability testing, and preparation for hybrid operation [8–10,13–17]. This guidance establishes the operational need for crypto agility, transition planning, hybrid operation, and interoperable algorithm replacement, but generally does not trace the algebraic dependencies of a specific KEM into individual policy and assurance controls [9,10,16,17].

Zero Trust architecture provides a complementary systems perspective based on explicit verification, least privilege, policy enforcement, continuous evaluation, and the rejection of implicit trust based solely on network location [3], while the NIST Cybersecurity Framework provides a broader governance and risk-management structure for maintaining such controls across the system lifecycle [18]. ML-KEM can support quantum-resistant key establishment within such an architecture, but it does not itself perform identity verification, authorization, device assessment, segmentation, or policy enforcement.

The research gap addressed here lies at the intersection of these bodies of work. Existing cryptographic sources rigorously define ML-KEM, while migration and Zero Trust guidance define organizational and architectural objectives. What remains insufficiently developed is a cross-layer mapping that explains how algebraic assumptions and standardized implementation requirements become inventory fields, admission criteria, negotiation policies, telemetry boundaries, migration states, rollback rules, and audit evidence. The framework developed in this paper supplies that mapping.

Analytical Method

The study applies a four-stage analytical method. First, the simplified algebraic relations underlying ML-KEM are decomposed to identify the operations and assumptions responsible for correctness intuition. Second, these relations are compared with the complete requirements of FIPS 203 to identify what the simplified model omits. Third, the resulting implementation dependencies are mapped to a Zero Trust threat model and crypto-agility lifecycle. Fourth, the framework is evaluated through assurance-to-control and threat-to-control traceability tables. The result is an architectural and analytical evaluation rather than a performance benchmark or a new cryptographic security proof.

3. Algebraic Setting: Quotient Rings, Modules, and Parameters

Fix a positive integer n along with an integer modulus $q \geq 2$ and then define the ring $R_q := \mathbb{Z}_q[X]/(X^n + 1)$. Thus, an element of R_q is a polynomial with coefficients in $\mathbb{Z}_q$, where addition and multiplication are performed modulo q on coefficients and modulo $X^n + 1$ on polynomials. In the Kyber-derived parameterization underlying ML-KEM, one takes $n = 256$ and $q = 3329$ [1]. Accordingly, every polynomial is represented modulo both the coefficient modulus q and the quotient ring relation $X^n \equiv -1$.

The choice of the polynomial $X^n + 1$ is mathematically and algorithmically significant. First, it endows the quotient with a negacyclic multiplication rule: whenever a term of degree at least n arises in a product, it is reduced using $X^n \equiv -1$, so the coefficient wraparound occurs with a sign change. This yields a particularly regular n -dimensional coefficient representation. Second, when n is a power of two, the polynomial $X^n + 1$ is closely related to the cyclotomic structures that are standard in ring- and module-lattice cryptography. This gives a ring that is both algebraically natural and highly amenable to fast multiplication algorithms, especially Number Theoretic Transform methods. In ML-KEM, the polynomial is therefore chosen not arbitrarily, but because it simultaneously provides compact representation, efficient arithmetic, and a structured setting compatible with the underlying Module-LWE hardness assumption.

For a fixed module rank k , let R_q^k denote the free left R_q -module of rank k , and let $R_q^{k \times k}$ denote the ring of k -by- k matrices with entries in R_q . Vectors are written as columns unless stated otherwise. If $A \in R_q^{k \times k}$ and $x \in R_q^k$, then $A \cdot x$ denotes the usual matrix-vector product over R_q . This notation allows ML-KEM to be viewed as linear algebra carried out over a structured quotient ring rather than over a field.

We further write A^T for the transpose of A . The public matrix satisfies $A \in R_q^{k \times k}$, while the secret and error vectors satisfy $s, e \in R_q^k$. Ephemeral vectors used during encapsulation are likewise elements of R_q^k , and message-embedding terms are ring elements in R_q . With this notation fixed, the algebraic identities in ML-KEM can be stated compactly and proved by ordinary distributivity in the module R_q^k .

4. Key Generation and Public Structure

4.1. Public and Secret Data

Key generation derives the public matrix $A \in R_q^{k \times k}$ deterministically from a public seed, while sampling the short secret and error vectors $s \in R_q^k$ and $e \in R_q^k$ [1]. The public vector $t \in R_q^k$ is then defined by the relation $t = A \cdot s + e$. Because e is small, the pair (A, t) may be regarded as a noisy linear relation in the unknown secret vector s . The security intuition is that recovering s from this perturbed relation is computationally hard, whereas a legitimate receiver can still exploit s algebraically during decapsulation.

4.2. Regeneration of the Public Matrix

In practice, the matrix A is not stored entry by entry as arbitrary data. Instead, it is deterministically regenerated from a seed. This design reduces key material and ensures that the same structured public matrix can be reconstructed whenever needed. Conceptually, however, it is helpful to think of A as an ordinary matrix over the ring R_q , since this makes the algebraic relationships transparent.

5. Simplified Algebraic Model of Encryption and Decryption

The relations in this section abstract the public-key encryption operations underlying ML-KEM so that the cancellation term and residual noise can be examined directly. The notation omits multiple mandatory elements of standardized ML-KEM, including byte-level encoding, compression and decompression, pseudorandom derivation, hashing,

shared-keying-material derivation, ciphertext validation, re-encryption checks, implicit rejection, and prescribed failure handling. Consequently, the equations below provide algebraic explanation only. All interoperable or security-critical implementations must follow FIPS 203 in full [1].

In the simplified encryption model, let $r \in R_q^k$ denote an ephemeral short vector, let $e_1 \in R_q^k$ denote an additional short error vector, and let $e_2 \in R_q$ denote a short error polynomial. These symbolic variables expose the algebraic structure of the underlying encryption relation; their generation in standardized ML-KEM is governed by the precise sampling and derivation procedures of FIPS 203 [1].

The first ciphertext component is defined by $u = A^T \cdot r + e_1$. The second ciphertext component is defined by $v = t^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m$, where $m \in \{0, 1\}^n \subset R_q$ is the embedded message representative. The ciphertext is therefore the pair $c = (u, v)$. The role of the term $\lceil q/2 \rceil \cdot m$ is to separate the two message values by a sufficient coefficient-wise margin so that coefficient-wise decoding in the simplified decryption model can recover the embedded representative provided the residual noise is small.

In the simplified decryption model, the receiver computes the intermediate element

$$w := v - s^T \cdot u \in R_q.$$

Substituting the ciphertext equations together with the public relation $t = A \cdot s + e$ yields

$$w = \lceil q/2 \rceil \cdot m + (e^T \cdot r + e_2 - s^T \cdot e_1).$$

The formal derivation of this identity is given in Appendix A. Thus, simplified decryption reduces to recovering the embedded binary message representative m from $w = \lceil q/2 \rceil \cdot m + (e^T \cdot r + e_2 - s^T \cdot e_1)$. If the residual noise term $e^T \cdot r + e_2 - s^T \cdot e_1$ remains within the decoding margin for each coefficient, coefficient-wise threshold decoding recovers the embedded binary message representative. This exact cancellation of the principal bilinear contribution is the algebraic mechanism that underlies correctness.

6. Quotient Ring Arithmetic

Before returning to the full parameter set, it is useful to examine a smaller quotient ring in which every algebraic step can be written explicitly. Consider the ring $R_{17} := \mathbb{Z}_{17}[X]/(X^2 + 1)$. Although this toy instance is not cryptographically realistic, it illustrates the same two-stage reduction that appears in ML-KEM: coefficient reduction modulo q and polynomial reduction modulo the defining relation of the quotient ring.

6.1. The Toy Ring and Its Reduction Map

Write $R_{17} := \mathbb{Z}_{17}[X]/(X^2 + 1)$. By definition, two polynomials in $\mathbb{Z}_{17}[X]$ determine the same element of R_{17} if and only if their difference is divisible by $X^2 + 1$. Equivalently, every computation in R_{17} may be carried out by first multiplying in $\mathbb{Z}_{17}[X]$, then reducing coefficients modulo 17, and finally replacing each occurrence of X^2 by -1 .

Proposition 1 (normal forms in R_{17}). *Every element of R_{17} admits a unique representative of the form $a + bX$ with $a, b \in \mathbb{Z}_{17}$. In particular, the quotient map $\mathbb{Z}_{17}[X] \rightarrow R_{17}$ is determined by the reduction rule $X^2 \equiv -1$.*

Proof. Since $X^2 + 1$ is monic of degree 2, the division algorithm in $\mathbb{Z}_{17}[X]$ shows that every polynomial $f(X)$ may be written uniquely as $f(X) = q(X)(X^2 + 1) + r(X)$, where either $r(X) = 0$ or $\deg r < 2$. Passing to the quotient identifies $f(X)$ with the remainder $r(X)$, so every class has a representative of degree at most 1. Uniqueness follows because if two

linear polynomials represent the same class, then their difference is divisible by $X^2 + 1$; but a nonzero polynomial of degree at most 1 cannot be divisible by a polynomial of degree 2. $\square$

Consider the residue classes of the polynomials $f(X) = 3X + 5$ and $g(X) = 2X + 4$ in R_{17} . We now compute their product rigorously as an element of the quotient ring.

6.2. Multiplication in the Quotient Ring

In the polynomial ring $\mathbb{Z}_{17}[X]$, one first computes

$$f(X)g(X) = (3X + 5)(2X + 4) = 6X^2 + 22X + 20.$$

Reducing coefficients modulo 17 yields

$$f(X)g(X) \equiv 6X^2 + 5X + 3.$$

Now pass to the quotient ring R_{17} . Since $X^2 + 1 = 0$ in R_{17} , one has $X^2 \equiv -1$, and therefore

$$f(X)g(X) \equiv 6(-1) + 5X + 3 = 5X - 3.$$

Finally, reducing the constant coefficient modulo 17 gives

$$f(X)g(X) \equiv 5X + 14 \text{ in } R_{17}.$$

Thus, the product of the residue classes of $f(X)$ and $g(X)$ is represented uniquely by the linear polynomial $14 + 5X$.

Proposition 2 (compatibility of coefficient and polynomial reduction). *Let $h(X) \in \mathbb{Z}_{17}[X]$. The residue class of $h(X)$ in R_{17} is unchanged whether one first reduces the coefficients of $h(X)$ modulo 17 and then imposes the relation $X^2 \equiv -1$, or first replaces each occurrence of X^2 by -1 and then reduces coefficients modulo 17. In both cases, one obtains the same element of R_{17} .*

Proof. By definition, $R_{17} = \mathbb{Z}_{17}[X]/(X^2 + 1)$, so computations in R_{17} are performed simultaneously modulo 17 on coefficients and modulo the ideal generated by $X^2 + 1$ on polynomials. Let $\pi: \mathbb{Z}_{17}[X] \rightarrow R_{17}$ denote the quotient map. Replacing X^2 with -1 corresponds exactly to passing to a polynomial congruent to $h(X)$ modulo the ideal $(X^2 + 1)$, while reducing integer coefficients modulo 17 is already built into the coefficient ring $\mathbb{Z}_{17}$. Since π depends only on the residue class of $h(X)$ modulo these two relations, either order of simplification produces the same image $\pi(h(X))$ in R_{17} . Therefore, coefficient reduction and polynomial reduction are compatible operations in the quotient ring. $\square$

Corollary 1 (well-definedness of multiplication on residue classes). *Let $f(X), f'(X), g(X), g'(X) \in \mathbb{Z}_{17}[X]$, and suppose $f(X) \equiv f'(X)$ and $g(X) \equiv g'(X)$ in R_{17} . Then $f(X)g(X) \equiv f'(X)g'(X)$ in R_{17} . In particular, multiplication of residue classes in R_{17} is well defined.*

Proof. The hypotheses mean that $f(X) - f'(X)$ and $g(X) - g'(X)$ both lie in the ideal generated by $X^2 + 1$. Equivalently, they represent the same residue classes under the quotient map $\pi: \mathbb{Z}_{17}[X] \rightarrow R_{17}$. Since π is a ring homomorphism, one has $\pi(f(X)g(X)) = \pi(f(X))\pi(g(X))$ and $\pi(f'(X)g'(X)) = \pi(f'(X))\pi(g'(X))$. Because $\pi(f(X)) = \pi(f'(X))$ and $\pi(g(X)) = \pi(g'(X))$, it follows that $\pi(f(X)g(X)) = \pi(f'(X)g'(X))$. Hence, $f(X)g(X)$ and $f'(X)g'(X)$ determine the same element of R_{17} . Therefore, multiplication is well defined on residue classes. $\square$

Soundness of computations in R_q : The Corollary explains why arithmetic in quotient rings is mathematically sound. Once one passes from polynomials to residue classes, the outcome of addition or multiplication no longer depends on which polynomial representatives are chosen, but only on the classes themselves. This is essential in ML-KEM: when one writes expressions such as $A \cdot s$, $A^T \cdot r$, or products of ring elements in R_q , one is not performing an ad hoc symbolic reduction, but a well-defined ring operation in the quotient $\mathbb{Z}_q[X]/(X^n + 1)$. The toy example therefore does more than illustrate a computational trick; it shows why the algebra used throughout ML-KEM is internally consistent and independent of presentation.

In the full ML-KEM setting, the same algebraic pattern occurs in the ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, with $n = 256$ and $q = 3329$ [1]. The essential difference is quantitative rather than structural: instead of linear polynomials over $\mathbb{Z}_{17}$, one works with degree-255 polynomials over $\mathbb{Z}_{3329}$, and polynomial products are evaluated efficiently by the Number Theoretic Transform. The toy example is therefore best understood as a minimal model of quotient ring multiplication, not as a faithful cryptographic parameter set. No security level, failure probability, implementation parameter, or performance conclusion for ML-KEM may be inferred from this toy construction. Its sole purpose is to make quotient-ring equivalence and negacyclic reduction visible in a small example.

7. Algebraic Interpretation

The mathematical significance of ML-KEM lies in the way it blends algebraic structure with controlled randomness. On the one hand, the quotient ring and module framework make the scheme highly structured and computationally efficient. On the other hand, the injected error terms prevent the public relation from revealing the secret vector. The resulting design is characteristic of modern lattice-based cryptography: efficient structured arithmetic is used not in spite of noise, but together with noise as an essential source of security.

8. Algebraic Lattice Perspective and Hardness Viewpoint

The algebraic core of ML-KEM may be understood as a module-lattice analogue of a noisy linear inversion problem. The public relation $t = A \cdot s + e$ is not merely an isolated equation in a quotient ring; rather, it defines a structured family of lattice points determined by the module R_q^k and the ring action of $R_q^{k \times k}$. When the secret and error vectors are sampled from suitably small distributions, the pair (A, t) may be viewed as a noisy module equation whose inversion is believed to be computationally hard. This is the algebraic content of the Module-LWE paradigm: depending on the formulation, Module-LWE asks an adversary either to recover hidden short information from noisy module equations or to distinguish such samples from uniformly random module elements.

More formally, if one samples a matrix $A \in R_q^{k \times k}$ from the prescribed public distribution and samples a short vector $s \in R_q^k$ together with a short error vector $e \in R_q^k$, then the distribution of pairs $(A, A \cdot s + e)$ is designed to be computationally indistinguishable from a distribution in which the second component is uniform in R_q^k , written $(A, A \cdot s + e) \approx_c (A, u)$ where u is uniform in R_q^k . In heuristic terms, the noise destroys exact linear recoverability while leaving enough algebraic structure for legitimate computation. This tension between structure and perturbation is the defining feature of lattice-based cryptography.

The term “module-lattice” reflects the fact that one may pass from ring elements to coefficient embeddings and thereby regard vectors in R_q^k as high-dimensional integer vectors endowed with additional algebraic structure. Under this viewpoint, multiplication by the public matrix A induces a structured linear transformation on a lattice-like ambient space, while the small vectors s , e , r , e_1 , and e_2 play the role of short perturbations. ML-KEM

thus occupies an intermediate position between unstructured LWE and fully ring-based constructions: it retains strong algebraic efficiency while avoiding the extreme specialization of a single-ring instance.

This algebraic analysis provides only one component of deployment assurance. FIPS 203 defines the complete ML-KEM, but security in an operational system additionally depends on faithful implementation and safe protocol composition. Relevant implementation obligations include correct sampling, approved entropy sources, exact encoding and compression behavior, constant-time processing, side-channel and fault resistance, ciphertext validation, implicit rejection, protection of secret-dependent intermediates, and use of tested or validated cryptographic providers [1,4–7].

These requirements form an implementation-assurance boundary. A system may negotiate the correct algorithm name and parameter set while still failing to preserve ML-KEM’s intended protections because of vulnerable implementation behavior [6,7], inadequate randomness [5], nonconforming failure behavior [4], protocol downgrade, unsafe hybrid composition [16,17], or leakage through telemetry. Mathematical validity, standards conformity, and operational approval must therefore be treated as related but distinct properties.

The deployment framework developed in the following sections converts these implementation dependencies into governance controls. For example, approved parameter sets become version-controlled policy objects; validated implementations become workload-admission criteria; library versions become inventory fields; negotiation results become observable compliance evidence; and implicit-rejection requirements constrain which failure details may be exposed to monitoring systems.

9. Boundary Between the Algebraic Model and Standardized ML-KEM

The simplified relations used in this paper isolate the noisy linear structure and cancellation identity, but standardized ML-KEM is a complete KEM rather than merely the displayed encryption relation. Table 1 identifies the principal distinction.

Table 1. Simplified algebraic model and standardized ML-KEM, based on FIPS 203 and NIST KEM guidance [1,4].

Simplified Algebraic Model	Complete Standardized ML-KEM	Deployment Significance
Represents short vectors and errors symbolically	Defines exact generation, derivation, and sampling procedures	Implementations must use conforming generation and approved entropy sources
Uses ring and module equations directly	Defines byte-level encoding, decoding, serialization, and parsing	Interoperability depends on exact representations, not only equivalent mathematics
Represents ciphertext as algebraic components (u) and (v)	Applies specified compression, decompression, encoding, and formatting	Compression effects and format validation influence correctness and interoperability
Uses message embedding and thresholding as correctness intuition	Executes the complete standardized decryption and decapsulation procedures	The pedagogical threshold model is not an implementation algorithm
Does not model all hashing operations	Specifies hashing and pseudorandom derivation functions	Approved function suites and domain separation are part of the standardized mechanism
Does not derive the final shared keying material	Defines derivation and binding of the resulting shared keying material	The KEM output is not merely the recovered message representative
Omits complete ciphertext validation	Includes prescribed processing and validation behavior	Failure behavior must not create a chosen-ciphertext or error oracle
Omits implicit rejection	Specifies implicit-rejection behavior	Monitoring must not expose secret-dependent distinctions between failure paths
Uses a sufficient coefficient-level noise margin	Relies on parameter-specific distributions and decryption-failure analysis	The (q/4) argument is explanatory rather than a complete probability analysis

9.1. Limits of the (q/4) Correctness Condition

The (q/4) condition used in the simplified derivation is a sufficient geometric separation condition. It explains why coefficients centered near 0 can be distinguished from coefficients centered near $\lceil q/2 \rceil$ when perturbations remain within disjoint neighborhoods. It should not be interpreted as a complete deterministic characterization of standardized ML-KEM correctness.

In full ML-KEM, residual behavior is influenced by sampled distributions, polynomial products, compression and decompression, encoding, parameter choices, and the complete decapsulation process. Formal decryption-failure analysis therefore concerns a probability distribution over these effects rather than only a deterministic bound on a single symbolic residual expression [11,12].

9.2. Parameter-Set Differences

ML-KEM-512, ML-KEM-768, and ML-KEM-1024 use different module ranks and provide different security, key-size, ciphertext-size, and performance profiles [1,11]. The quotient-ring dimension and coefficient modulus remain standardized, but the parameter sets are not operationally interchangeable. Selection of a parameter set must therefore be a version-controlled policy decision based on required security strength, platform constraints, protocol support, implementation validation, and interoperability requirements [1,4].

9.3. Implementation and Composition Risks

A secure deployment must address risks outside the algebraic derivation, including non-constant-time behavior, microarchitectural and physical side channels, fault injection, inadequate randomness, vulnerable library versions, incorrect serialization, and unsafe failure reporting [4–7]. Protocol-level risks include downgrade to legacy key establishment, algorithm confusion, incorrect hybrid combination, insufficient binding of negotiation results, and fallback to unauthorized profiles [9,16,17]. These risks are not repaired by the mathematical strength of Module-LWE and must instead be controlled through implementation assurance, protocol design, and system governance [4,9,18].

10. ML-KEM Deployment in a Crypto-Agile Zero Trust Architecture

Zero Trust is an architectural approach in which access is based on explicit verification, policy, identity, device or workload state, least privilege, and continuous evaluation rather than assumed trust derived from network location [3], with broader governance, risk ownership, and lifecycle management informed by the NIST Cybersecurity Framework [18]. Within this architecture, ML-KEM may provide post-quantum key establishment, but it does not independently provide identity, authorization, segmentation, endpoint integrity, or policy enforcement. Operational deployments will typically pair ML-KEM with standardized post-quantum digital signatures, such as ML-DSA defined in FIPS 204, as complementary cryptographic services within approved cryptographic profiles [1,19].

10.1. Threat Model

The deployment framework considers the following threats:

1. An adversary records protected traffic for future quantum-enabled cryptanalysis [10,14,15].
2. An intermediary attempts to downgrade peers to an unauthorized traditional or weaker profile [9,17].
3. A workload continues using a deprecated algorithm, parameter set, or library version.
4. A compromised or misconfigured endpoint presents an approved algorithm name while using a nonconforming implementation.

5. Weak randomness, side-channel leakage, fault behavior, or unsafe error reporting undermines the implementation [4–7].
6. A hybrid deployment combines algorithms incorrectly or fails to bind negotiation results to the protocol transcript [16,17].
7. A migration or rollback mechanism returns a service to an unauthorized configuration.
8. Monitoring telemetry is incomplete, manipulated, or interpreted incorrectly.
9. An automated governance component recommends an unsafe operational change [20].

The threat categories synthesize risks identified in KEM guidance, PQC transition and crypto-agility guidance, hybrid-protocol specifications, implementation-security research, Zero Trust guidance, and AI risk-management guidance [3–7,9,10,16–18,20]. The framework distinguishes between threats addressed by ML-KEM and threats addressed by surrounding controls. ML-KEM addresses quantum-resistant key establishment under its specified assumptions. Identity controls address unauthorized principals; implementation controls address leakage and nonconformance; protocol controls address downgrade and composition; and governance controls address inventory, migration, monitoring, and audit.

10.2. Functional Architecture

The use of policy decision and enforcement functions follows the logical Zero Trust model of NIST SP 800-207, while the inventory, lifecycle, and governance functions extend that model using current crypto-agility and cybersecurity-risk guidance [3,9,18]. The architecture is illustrated in Figure 1. As shown in Figure 1, the architecture integrates Zero Trust policy decision and enforcement functions with approved cryptographic services, negotiation controls, inventory, telemetry, and lifecycle governance. The principal components are summarized below:

- Identity and posture source: This establishes the identity and assessed state of the user, workload, service, or device.
- Policy decision point: This determines which cryptographic profiles are authorized for a specific transaction or workload.
- Policy enforcement point: This permits, denies, isolates, or restricts a connection according to the approved decision.
- Approved cryptographic provider: This is a software, hardware, or service component authorized by deployment policy to execute the complete standardized ML-KEM. Approval may include implementation testing, platform qualification, and formal validation where required.
- Negotiation controller: This restricts negotiation to approved traditional, hybrid, or post-quantum profiles and prevents unauthorized downgrade.
- Cryptographic inventory: This records algorithms, parameter sets, certificates, protocol versions, providers, library versions, dependencies, and deployment status.
- Telemetry and conformance service: This compares observed non-secret behavior with approved policy.
- Migration and rollback controller: This stages approved changes and returns services only to the last authorized secure configuration.
- Audit repository: This preserves policy versions, negotiation evidence, exceptions, migration decisions, and administrative actions.

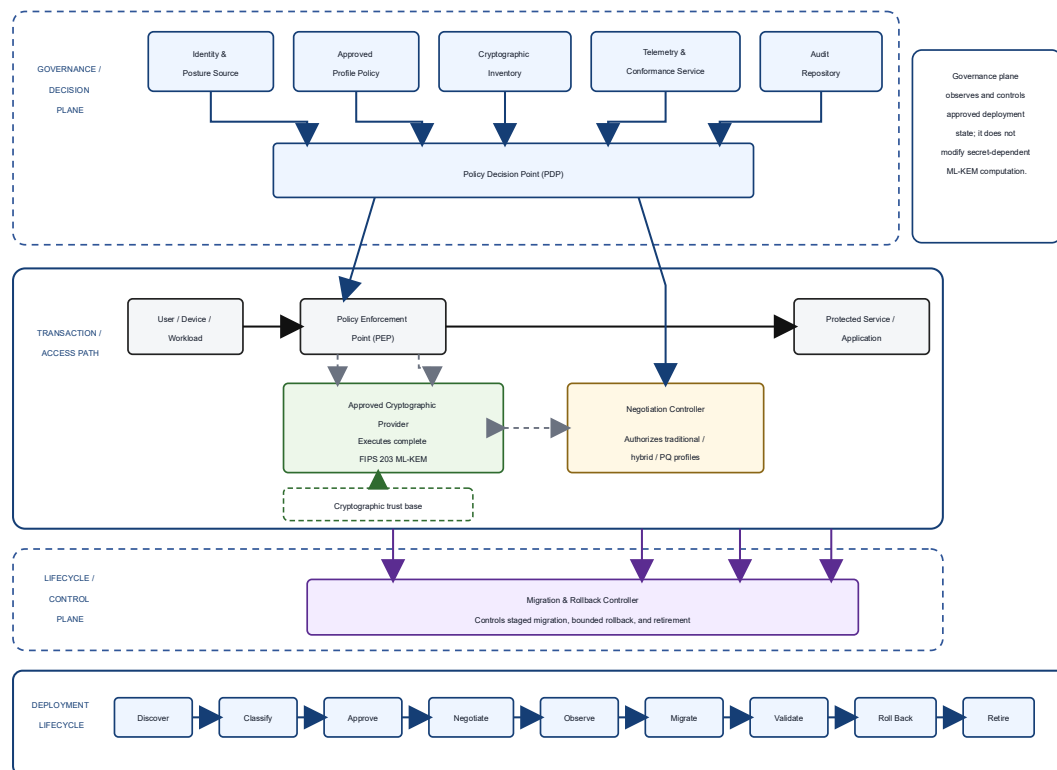


Figure 1. ML-KEM Zero Trust deployment-assurance architecture, informed by NIST Zero Trust, KEM, and crypto-agility guidance [3,4,9]. Solid arrows denote operational processing and message flow between architectural components. Vertical arrows denote policy, governance, and lifecycle-control relationships. Dashed boundaries delineate architectural control domains rather than communication paths.

10.3. Crypto-Agility Lifecycle

The lifecycle is the synthesis of NIST transition, crypto-agility, and cybersecurity-governance guidance [9,10,18]. Crypto agility is represented as a controlled lifecycle. Table 2 summarizes the principal lifecycle stages, their required activities, and representative evidence or control mechanisms derived from current NIST crypto-agility and transition guidance.

Table 2. Crypto-agility lifecycle, informed by NIST crypto-agility and transition guidance [9,10].

Stage	Required Activity	Evidence or Control
Discover	Identify algorithms, protocols, libraries, certificates, parameter sets, and dependencies	Cryptographic bill of materials and runtime inventory
Classify	Determine sensitivity, lifetime, quantum exposure, and interoperability needs	Data and workload classification
Approve	Define permitted traditional, hybrid, and post-quantum profiles	Version-controlled cryptographic policy
Negotiate	Select only mutually supported and authorized profiles	Authenticated negotiation and downgrade protection
Observe	Record approved non-secret runtime attributes	Algorithm, parameter set, protocol, provider, and version telemetry
Migrate	Introduce approved profiles through staged deployment	Change control, canary deployment, and compatibility testing
Validate	Confirm interoperability, conformance, and policy compliance	Test results, attestation, and audit evidence
Roll Back	Return to the last approved secure profile when required	Bounded and authorized rollback procedure
Retire	Disable and remove deprecated mechanisms	Updated inventory and enforcement policy

10.4. Protocol and Policy Workflow

A representative transaction proceeds as follows:

1. The initiating workload presents identity and available posture evidence.
2. The policy decision point evaluates identity, workload sensitivity, platform state, approved providers, and current migration policy.
3. The negotiation controller permits only an authorized traditional, post-quantum traditional hybrid, or post-quantum profile [9,16,17].
4. The approved provider executes the complete standardized mechanism rather than the simplified equations presented for explanation [1,4].
5. The enforcement point permits the protected session only when identity, posture, negotiation, and cryptographic-provider requirements are satisfied.
6. Non-secret telemetry records the negotiated profile, parameter set, protocol version, provider, library version, policy identifier, and conformance result [3,9,18] while excluding secret keys, shared keying material, randomness, and secret-dependent intermediate values.
7. Governance services detect unauthorized fallback, deprecated components, configuration drift, interoperability failure, or incomplete migration.
8. Responses are limited to alerting, denial, isolation, staged remediation, or rollback to a previously approved profile [9,18].
9. Policy and migration decisions are written to an auditable repository.

The implementation requirements associated with the representative workflow are summarized in Table 3, which maps the principal ML-KEM implementation dependencies to corresponding deployment controls and Zero Trust governance mechanisms.

Table 3. Mapping ML-KEM dependencies to deployment controls, synthesized from FIPS 203, NIST KEM guidance, Zero Trust architecture, and crypto-agility guidance [1,3,4,9].

Cryptographic Dependency	Implementation Requirement	Zero Trust or Governance Control
Approved ML-KEM parameter set	Exact conforming implementation	Version-controlled allowed-profile policy
Correct sampling and derivation	Approved entropy and generation procedures [5]	Provider admission, validation evidence, and health monitoring
Exact encoding and compression	Standards-conformant serialization and parsing	Interoperability and conformance testing
Ciphertext validation and implicit rejection	Uniform and failure behavior that does not reveal validity-dependent distinctions	Restricted telemetry and failure-detail policy
Constant-time secret processing	Side-channel-aware implementation	Platform and library admission criteria
Complete shared-keying-material derivation	Approved hashing and KDF behavior	Protocol-profile and implementation-version control
Hybrid composition	Approved combiner and transcript binding [16,17]	Negotiation policy and downgrade prevention
Migration and rollback	Reversible, tested configuration changes	Staged deployment and last-approved-state recovery

10.5. Worked Migration Scenario

Consider a service population initially using an approved traditional key-establishment profile. During the first migration stage, cryptographic inventory identifies endpoints, libraries, and dependent protocols that can support a hybrid traditional–ML-KEM profile. This is a post-quantum traditional hybrid profile combining a traditional key-establishment mechanism with ML-KEM [16,17]. The hybrid profile is introduced through canary deployment and is permitted only when both peers use approved providers and protocol

versions. Negotiation results and compatibility failures are observed without collecting secret values [9,17].

Services that pass interoperability and conformance checks advance to broader hybrid deployment. Services that fail remain on the last approved traditional profile only when current policy still authorizes that profile; otherwise, they are isolated or placed under an exception process. A later policy version may designate a post-quantum-preferred profile. Rollback from that state is permitted only to a named, previously approved hybrid profile, not to any profile that happens to restore connectivity. This prevents crypto agility from becoming uncontrolled downgrade capability [9,10,18].

11. Bounded AI-Assisted Cryptographic Governance

AI-assisted monitoring may support cryptographic inventory, anomaly detection, migration planning, and operational correlation, but it does not create cryptographic assurance. Consistent with the NIST AI Risk Management Framework, its role, authority, inputs, outputs, oversight, and failure controls must be explicitly governed and evaluated [20]. The security properties of ML-KEM arise from the standardized design, its assumptions, and faithful implementation [1,4]. An AI component therefore belongs to the governance and observability layer and must remain outside secret-dependent cryptographic processing.

11.1. Permitted Observations

Subject to data-minimization, access-control, documentation, measurement, and monitoring requirements established for deployment [18,20], an AI-assisted governance service may process non-secret operational information such as:

- Negotiated algorithm and profile identifiers;
- ML-KEM parameter-set identifiers;
- Protocol and provider versions;
- Cryptographic-library version and validation status;
- Certificate and public-key metadata;
- Workload identity and attestation state;
- Approved-policy identifiers;
- Cryptographic inventory records;
- Configuration drift;
- Migration-stage status;
- Interoperability test results;
- Aggregated, non-secret operational health indicators.

It must not collect or infer:

- Secret keys;
- Shared secrets;
- Encapsulation randomness;
- Secret-dependent decapsulation intermediates;
- Protected plaintext;
- Sensitive process memory;
- Detailed failure distinctions that could create a timing, error-message, or response-distinguishability oracle.

11.2. Permitted and Prohibited Actions

High-impact actions remain subject to deterministic policy enforcement or human authorization. The AI service may recommend an action or initiate an approved workflow, but the policy decision point remains authoritative. This separation preserves deterministic enforcement of approved cryptographic policy and provides a defined human-oversight

boundary for high-impact automated recommendations [3,20]. Table 4 summarizes representative governance actions that may be recommended or coordinated by the AI service, together with cryptographic actions that remain prohibited and require deterministic enforcement under the proposed architecture.

Table 4. Authority boundary for AI-assisted cryptographic governance, informed by the NIST AI RMF [20].

Permitted Governance Actions	Prohibited Cryptographic Actions [1,4,6,7,20]
Detect use of a deprecated or unauthorized profile	Create or modify a cryptographic algorithm
Correlate inventory with observed runtime configuration	Generate, alter, export, or inspect secret keys
Flag suspected downgrade or configuration drift	Select arbitrary or unapproved parameter sets
Identify endpoints that have not completed migration	Modify ML-KEM arithmetic or standardized processing
Recommend staged remediation	Bypass ciphertext validation or implicit rejection
Open a review, change, or exception workflow	Suppress or alter mandatory failure handling
Initiate a pre-approved containment workflow	Invent an unapproved hybrid combination
Recommend rollback to a named approved profile	Authorize fallback to a prohibited legacy mechanism

11.3. Error and False-Positive Controls

AI-assisted recommendations are controlled through a Govern–Map–Measure–Manage risk process adapted from the NIST AI RMF and its Playbook [20]:

1. Deterministic validation against current cryptographic policy;
2. Confidence thresholds appropriate to the effect of the proposed action;
3. Corroboration across independent telemetry sources;
4. Human approval for high-impact migration, isolation, or rollback;
5. Canary deployment and staged rollout;
6. Reversible configuration changes;
7. Immutable logging of inputs, recommendations, approvals, and outcomes;
8. Retention of the last approved secure state;
9. Periodic testing for drift, false positives, and unsafe recommendations.

11.4. Assurance Versus Governance

Cryptographic assurance concerns the properties supported by the algorithm, its security assumptions, the standard, and a faithful implementation. Implementation assurance concerns whether those requirements survive realization in software, hardware, and protocol composition. Governance assurance concerns whether approved mechanisms are selected, deployed, monitored, migrated, and audited correctly. AI assistance may strengthen governance assurance by improving visibility and coordination, but it cannot repair defective cryptography or convert a nonconforming implementation into a secure one [4,9,18,20].

12. Results and Discussion

The analysis produces a three-layer assurance model in which algebraic assurance, implementation assurance, and governance assurance are related but noninterchangeable. Algebraic cancellation explains why recovery is possible in the simplified model. Standards conformity determines whether the complete ML-KEM is realized correctly [1,4]. Zero Trust governance determines whether an approved realization is selected, negotiated, monitored, migrated, and maintained in operation.

The central result is the traceability relationship shown in Table 3. Mathematical and standardized dependencies become implementation requirements, while implementation requirements become policy objects, admission criteria, inventory fields, telemetry restric-

tions, migration states, and audit evidence. This relationship makes the role of the algebraic exposition operationally relevant without implying that Zero Trust changes the underlying cryptographic primitive.

The architecture also establishes a clear boundary for automation. Governance systems may reason about approved profiles and non-secret operational state, but they must not alter cryptographic processing or consume secret-dependent data. This boundary is particularly important when AI-assisted components are introduced, because an unconstrained orchestration mechanism could otherwise weaken downgrade protections, expose failure distinctions, or authorize unsafe rollback [4,17,20].

Summary of Findings

Table 5 summarizes the principal analytical findings, their technical meaning, and the corresponding operational consequences for secure ML-KEM deployment.

Table 5. Principal findings.

Finding	Technical Meaning	Operational Consequence
Algebraic cancellation is necessary but not sufficient for deployment assurance	The simplified identity explains correctness intuition but omits mandatory standardized processing	Implementers must use the complete FIPS 203 mechanism
Mathematical, implementation, and governance assurance are distinct	A valid primitive can still be deployed insecurely	Approval must include provider, platform, protocol, and policy evidence
Crypto agility is a controlled lifecycle	Replaceability without governance can become downgrade capability	Migration and rollback must use versioned approved profiles
Zero Trust surrounds but does not redefine ML-KEM	ML-KEM provides key establishment, not identity or authorization	Identity, posture, policy, and enforcement remain separate architectural functions
AI can support governance but not cryptographic assurance	AI may process non-secret configuration and policy evidence	Cryptographic computations and secret-dependent data remain outside AI authority

13. Limitations

This study is analytical and architectural. It does not present a new cryptographic primitive, security reduction, implementation, interoperability test campaign, side-channel evaluation, or performance benchmark. The simplified equations intentionally omit mandatory portions of FIPS 203 and are not suitable as implementation guidance. The proposed architecture must also be adapted to the specific identity system, transport protocol, cryptographic provider, assurance regime, and operational environment in which it is deployed.

The threat-to-control mappings demonstrate design coverage rather than empirical control effectiveness. Future work should implement the architecture in a representative service environment, measure migration and negotiation behavior, evaluate telemetry sufficiency, test downgrade and rollback controls, and assess the error characteristics of AI-assisted recommendations.

14. Conclusions

ML-KEM provides a standardized post-quantum key-establishment mechanism whose underlying algebra can be understood through module operations, quotient-ring arithmetic, controlled noise, and cancellation of a principal bilinear term. That explanation is valuable, but it is not the complete mechanism and does not by itself establish deployment security.

Operational assurance requires the complete FIPS 203 process, a conforming and side-channel-aware implementation, safe protocol composition, controlled negotiation, accurate cryptographic inventory, and governed migration [1,4–7,9,17]. Within a Zero

Trust architecture, these requirements can be represented as explicit policy, admission, monitoring, rollback, and audit controls.

The principal contribution of this paper is the connection among these layers. By tracing mathematical and standardized dependencies into implementation obligations and governance controls, the framework provides a clearer path from post-quantum cryptographic theory to controlled operational adoption. AI-assisted monitoring may improve visibility and migration coordination within this framework, but only when its observations and authority remain bounded outside the cryptographic trust base.

Author Contributions: Conceptualization, W.E. and M.V.; methodology, W.E. and M.V.; validation, M.V. and J.W.; formal analysis, W.E. and J.W.; investigation, W.E., M.V. and J.W.; resources, M.V. and J.W.; data curation, W.E.; writing—original draft preparation, W.E., M.V. and J.W.; writing—review and editing, J.W.; visualization, J.W.; supervision, J.W.; project administration, J.W.; funding acquisition, W.E., M.V. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author William Edwards was employed by the company Deep Blue Krypto Group. Authors Miroslav Vukovic and Jeffrey Wallace were employed by the company Black Swan Syndicate LLC. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Glossary

Term	Definition
Quotient ring	A ring formed by identifying polynomials that differ by a multiple of a fixed modulus polynomial, here $X^n + 1$.
Negacyclic reduction	The rule $X^n \equiv -1$, which causes coefficient wraparound with a sign change when polynomials are multiplied in R_q .
Module	An algebraic structure analogous to a vector space, but defined over a ring such as R_q rather than over a field.
Cryptographic profile	A version-controlled specification of the approved key-establishment mechanism or combination, parameter set, protocol binding, cryptographic provider requirements, implementation versions, negotiation rules, and associated security policy.
Public relation	The noisy linear equation $t = A \cdot s + e$ used in key generation, in which the error term e hides the exact linear relation.
Ciphertext components	The pair (u, v) produced during encapsulation, in which both components combine public data, fresh randomness, and message embedding.
Residual noise	The error term $e^T \cdot r + e_2 - s^T \cdot e_1$ that remains after cancellation during decapsulation.
Decapsulation	The process of recovering the embedded message representative by computing $w = v - s^T \cdot u$ and thresholding the resulting coefficients.
Module-LWE	The hardness assumption underlying ML-KEM, according to which noisy linear relations over modules are computationally difficult to solve.
Normal form	A canonical representative of a residue class in a quotient ring, such as $a + bX$ in R_{17} .
Thresholding	The coefficient-wise decoding rule that decides whether a coefficient is closer to 0 or to $\lceil q/2 \rceil$.
Cryptographic governance	The set of processes and controls used to approve, deploy, monitor, migrate, retire, and audit cryptographic mechanisms.

Cryptographic trust base	The set of implementation components whose correct and secure operation is required to preserve the intended cryptographic properties, including the approved provider, entropy source, key handling, and security-critical platform dependencies.
Governance assurance	The confidence, supported by evidence, that those cryptographic governance processes and controls operate as intended.
Algebraic assurance	Confidence derived from the correct statement and interpretation of the algebraic construction, parameter domain, cancellation relation, and sufficient correctness conditions used in the explanatory model.

Appendix A. Formal Algebraic Restatement

This appendix restates the core algebra of ML-KEM in a notation consistent with the main text. Throughout, let $R_q := \mathbb{Z}_q[X]/(X^n + 1)$, let k be the module rank, and write R_q^k and $R_q^{k \times k}$ for the free module of rank k and the ring of k -by- k matrices over R_q , respectively. Vectors are treated as column vectors, and transposition converts them to row vectors in the evident way.

Appendix A.1. Definition and Notation

Definition A1 (ring and module setting). Let $R_q := \mathbb{Z}_q[X]/(X^n + 1)$. For a fixed module rank k , let R_q^k denote the free left R_q -module of rank k , and let $R_q^{k \times k}$ denote the set of k -by- k matrices with entries in R_q . If $A \in R_q^{k \times k}$ and $x \in R_q^k$, then $A \cdot x$ denotes the usual matrix-vector product over R_q . The transpose A^T is defined entrywise, and if x is a column vector in R_q^k , then x^T is the corresponding row vector.

For the Kyber-derived parameterization underlying ML-KEM, one takes $n = 256$ and $q = 3329$ [1]. The public matrix satisfies $A \in R_q^{k \times k}$, the secret and error vectors satisfy $s, e \in R_q^k$, and the ephemeral encapsulation data satisfy $r, e_1 \in R_q^k$ and $e_2, m \in R_q$.

Symbol	Meaning
R_q	The quotient ring $\mathbb{Z}_q[X]/(X^n + 1)$.
A	The public matrix in $R_q^{k \times k}$.
s	The short secret vector in R_q^k .
e	The key-generation error vector in R_q^k .
t	The public vector defined by $t = A \cdot s + e$.
r	The ephemeral vector used during encapsulation, in R_q^k .
e_1	The encapsulation noise vector in R_q^k .
e_2	The encapsulation error polynomial, or ring element, in R_q .
u	The first ciphertext component, defined by $u = A^T \cdot r + e_1$.
v	The second ciphertext component, defined by $v = t^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m$.
w	The decapsulation intermediate defined by $w = v - s^T \cdot u$.
m	The embedded message representative in R_q .

Appendix A.2. Public Relation

Definition A2 (public relation). Key generation deterministically derives the public matrix $A \in R_q^{k \times k}$ from a public seed and samples the short secret and error vectors $s \in R_q^k$ and $e \in R_q^k$, and defines the public vector $t \in R_q^k$ by

$$t = A \cdot s + e.$$

The pair (A, t) is therefore a noisy linear relation in the unknown vector s . The qualitative security intuition is that the perturbation by e hides the exact solution while preserving enough structure for legitimate decapsulation.

Appendix A.3. Ciphertext Formation

Definition A3 (ciphertext components). *Given the public data, encapsulation samples r , $e_1 \in R_q^k$ and $e_2 \in R_q$, and forms the ciphertext components [1]*

$$\begin{aligned} u &= A^T \cdot r + e_1, \\ v &= t^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m. \end{aligned}$$

The ciphertext is $c = (u, v)$. The term $\lceil q/2 \rceil \cdot m$ embeds the message into coefficient space so that decoding can distinguish the two message values as long as the residual error remains sufficiently small.

Appendix A.4. Correctness Proposition

Proposition A1 (algebraic form of the simplified decryption relation). *Let $u = A^T \cdot r + e_1$ and $v = t^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m$, where $t = A \cdot s + e$. If simplified decryption defines $w := v - s^T \cdot u$, then [1,11]*

$$w = \lceil q/2 \rceil \cdot m + (e^T \cdot r + e_2 - s^T \cdot e_1).$$

Proof. By definition of the simplified decryption relation, one sets $w := v - s^T \cdot u$. We now substitute the definitions of u and v from Definition A.3. Since $u = A^T \cdot r + e_1$ and $v = t^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m$, it follows that

$$w = (t^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m) - s^T \cdot (A^T \cdot r + e_1).$$

Next, use the public relation $t = A \cdot s + e$ from Definition A.2. Taking transposes yields

$$t^T = (A \cdot s + e)^T = s^T \cdot A^T + e^T,$$

where the second equality uses the elementary transpose identities $(A \cdot s)^T = s^T \cdot A^T$ and $(x + y)^T = x^T + y^T$. $\square$

Substituting this expression for t^T into the formula for w gives

$$w = ((s^T \cdot A^T + e^T) \cdot r + e_2 + \lceil q/2 \rceil \cdot m) - s^T \cdot (A^T \cdot r + e_1).$$

Now distribute multiplication over addition in R_q and expand both products:

$$w = s^T \cdot A^T \cdot r + e^T \cdot r + e_2 + \lceil q/2 \rceil \cdot m - s^T \cdot A^T \cdot r - s^T \cdot e_1.$$

The term $s^T \cdot A^T \cdot r$ appears once with a positive sign and once with a negative sign, so these two contributions cancel exactly. Therefore,

$$w = \lceil q/2 \rceil \cdot m + (e^T \cdot r + e_2 - s^T \cdot e_1).$$

This is precisely the claimed identity. It shows that the simplified decryption result separates into two parts: the embedded message term $\lceil q/2 \rceil \cdot m$ and the residual noise term $e^T \cdot r + e_2 - s^T \cdot e_1$. The proposition therefore isolates the exact algebraic cancellation responsible for correctness.

Corollary A1 (correctness under bounded coefficient-wise noise). *Let $m \in \{0, 1\}^n \subset R_q$, and let $\eta := e^T \cdot r + e_2 - s^T \cdot e_1 \in R_q$. Suppose that, for every coefficient, the representative of the corresponding coefficient of η in the centered interval $[-(q-1)/2, (q-1)/2]$ has absolute value strictly less than $q/4$ [12]. Then coefficient-wise thresholding of w recovers the embedded message representative m [1,11,12].*

Proof. By Proposition A.1, one has $w = \lceil q/2 \rceil \cdot m + \eta$. Let w_i , m_i , and η_i denote the i -th coefficients of w , m , and η , respectively. Then, coefficient-wise, the preceding identity becomes $w_i = \lceil q/2 \rceil \cdot m_i + \eta_i$ in $\mathbb{Z}_q$. By hypothesis, the centered representative of η_i has absolute value strictly less than $q/4$ for every i . We now consider the two possible values of m_i .

First, suppose $m_i = 0$. Then $w_i = \eta_i$ in $\mathbb{Z}_q$. Since the centered representative of η_i has absolute value less than $q/4$, the coefficient w_i lies within distance $q/4$ of 0. In particular, w_i cannot lie in the decoding region centered at $\lceil q/2 \rceil$.

Second, suppose $m_i = 1$. Then $w_i = \lceil q/2 \rceil + \eta_i$ in $\mathbb{Z}_q$. Again, using the bound on η_i , the coefficient w_i lies within distance $q/4$ of $\lceil q/2 \rceil$ in centered representation around that value. Consequently, w_i cannot lie in the decoding region centered at 0.

The two decoding regions are disjoint because their centers, namely 0 and $\lceil q/2 \rceil$, are separated by approximately $q/2$, whereas each permitted perturbation has radius strictly less than $q/4$. Therefore, for each coefficient i , coefficient-wise thresholding correctly determines whether m_i equals 0 or 1. Since this holds for every coefficient, thresholding recovers the entire embedded message representative m .

Scope and limitation of the appendix: Proposition A.1 and Corollary A.1 formalize the cancellation identity and a sufficient coefficient-level decoding condition for the simplified algebraic model. They do not reproduce the full FIPS 203 mechanism or its parameter-specific decryption-failure analysis. Standardized sampling, compression, encoding, hashing, key derivation, ciphertext validation, implicit rejection, and complete decapsulation behavior remain governed exclusively by FIPS 203 and the supporting analyses [1,11,12]. $\square$

References

1. FIPS 203; Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST (National Institute of Standards and Technology) Computer Security Resource Center: Gaithersburg, MD, USA, 2024. Available online: <https://csrc.nist.gov/pubs/fips/203/final> (accessed on 15 June 2026).
2. National Institute of Standards and Technology. *Post-Quantum Cryptography Standardization Process*; NIST Computer Security Resource Center: Gaithersburg, MD, USA. Available online: <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization> (accessed on 15 June 2026).
3. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; NIST Special Publication 800-207; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. Available online: <https://csrc.nist.gov/pubs/sp/800/207/final> (accessed on 15 June 2026).
4. Alagic, G.; Barker, E.; Chen, L.; Moody, D.; Robinson, A.; Silberg, H.; Waller, N. *Recommendations for Key-Encapsulation Mechanisms*; NIST Special Publication 800-227; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025. [CrossRef]
5. Sönmez Turan, M.; Barker, E.; Kelsey, J.; McKay, K.; Baish, M.; Boyle, M. *Recommendation for the Entropy Sources Used for Random Bit Generation*; NIST Special Publication 800-90B; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018. [CrossRef]

6. Chen, B.; Wang, Y.; Shome, P.; Fletcher, C.; Kohlbrenner, D.; Paccagnella, R.; Genkin, D. GoFetch: Breaking Constant-Time Cryptographic Implementations Using Data Memory-Dependent Prefetchers. In Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, USA, 14 August 2024; pp. 1117–1134.
7. Bernstein, D.J.; Schwabe, P. KyberSlash: Exploiting Secret-Dependent Division Timings in Kyber Implementations; Report 2024/1049; Cryptology ePrint Archive. 2024. Available online: <https://eprint.iacr.org/2024/1049> (accessed on 21 July 2026).
8. Souppaya, M.; Polk, T.; Barker, W. *Getting Ready for Post-Quantum Cryptography: Exploring Challenges Associated with Adopting and Using Post-Quantum Cryptographic Algorithms*; NIST Cybersecurity White Paper; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021. Available online: <https://www.nist.gov/publications/getting-ready-post-quantum-cryptography-exploring-challenges-associated-adopting-and> (accessed on 15 June 2026).
9. Barker, E.; Chen, L.; Cooper, D.; Moody, D.; Regenscheid, A.; Souppaya, M.; Newhouse, W.; Housley, R.; Turner, S.; Barker, W.; et al. *Considerations for Achieving Crypto Agility: Strategies and Practices*; NIST Cybersecurity White Paper 39, Update 1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2026. [CrossRef]
10. Moody, D.; Perlner, R.; Regenscheid, A.; Robinson, A.; Cooper, D. *Transition to Post-Quantum Cryptography Standards, NIST Interagency Report 8547, Initial Public Draft*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024.
11. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation, Version 3.01. 2021. Available online: <https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf> (accessed on 15 June 2026).
12. Fang, B.; Wang, W.; Zhao, Y. Tight Analysis of Decryption Failure Probability of Kyber in Reality; Cryptology ePrint Archive Paper 2022/212, 2022. Available online: <https://eprint.iacr.org/2022/212.pdf> (accessed on 2 May 2026).
13. National Institute of Standards and Technology. *Post-Quantum Cryptography Project*; NIST Computer Security Resource Center: Gaithersburg, MD, USA. Available online: <https://csrc.nist.gov/projects/post-quantum-cryptography> (accessed on 15 June 2026).
14. Cybersecurity and Infrastructure Security Agency; National Security Agency; National Institute of Standards and Technology. Quantum-Readiness: Migration to Post-Quantum Cryptography. 2023. Available online: <https://www.cisa.gov/resources-tools/resources/quantum-readiness-migration-post-quantum-cryptography> (accessed on 15 June 2026).
15. National Security Agency. Post-Quantum Cryptography: CISA, NIST, and NSA Recommend How to Prepare Now. 2023. Available online: <https://www.nsa.gov/Press-Room/Press-Releases-Statements/Press-Release-View/Article/3498776/post-quantum-cryptography-cisa-nist-and-nsa-recommend-how-to-prepare-now/> (accessed on 15 June 2026).
16. Driscoll, M.; Fluhrer, S.; Ghosh, S.; Stebila, D. *Terminology for Post-Quantum Traditional Hybrid Schemes*, RFC 9794; Internet Engineering Task Force: Fremont, FL, USA, 2025. [CrossRef]
17. Stebila, D.; Fluhrer, S.; Gueron, S. *Hybrid Key Exchange in TLS 1.3*, RFC 9954; Internet Engineering Task Force: Fremont, FL, USA, 2026. [CrossRef]
18. Pascoe, C.; Quinn, S.; Scarfone, K. *The NIST Cybersecurity Framework (CSF) 2.0*; NIST Cybersecurity White Paper 29; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024. [CrossRef]
19. *FIPS 204; Module-Lattice-Based Digital Signature Standard*. NIST (National Institute of Standards and Technology) Computer Security Resource Center: Gaithersburg, MD, USA, 2024. Available online: <https://csrc.nist.gov/pubs/fips/204/final> (accessed on 2 May 2026).
20. Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.